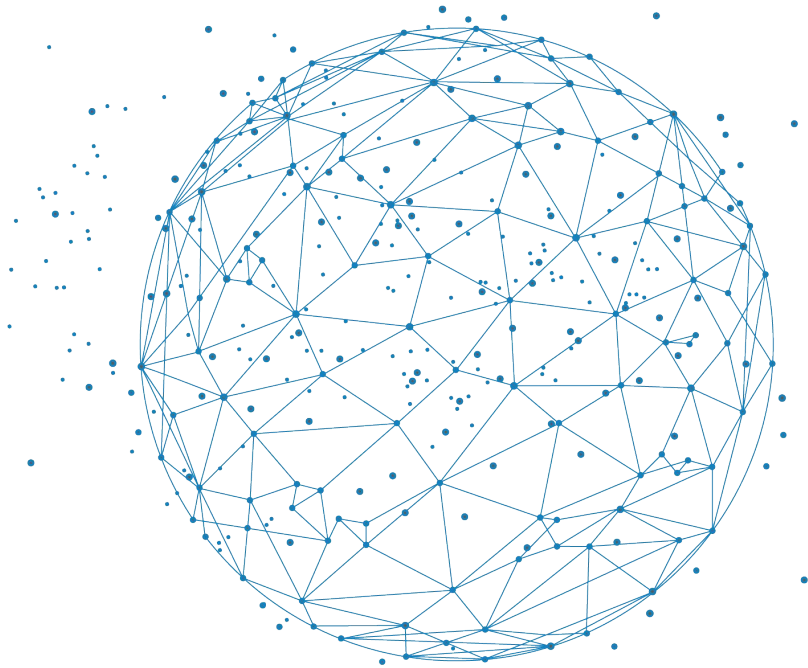


2025 June 6th



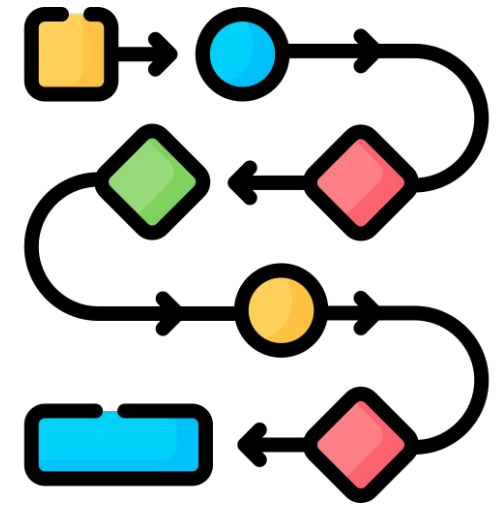
Introduction to FAIR software environnement Conda, Docker and Apptainer

Julien Seiler

In bioinformatics, **reproducibility** refers to the ability to replicate the results of an experiment or analysis using the same data and computational steps.

Challenges in Achieving Reproducibility

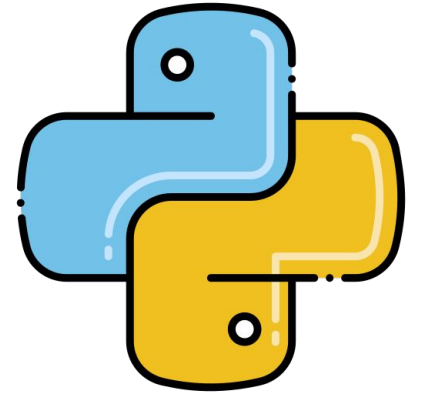
- **Complexity of Workflows:** Bioinformatics workflows often involve multiple steps, **tools**, and data sources, making them complex and difficult to replicate.
- **Dependency Management:** Different software versions, libraries, and dependencies can lead to inconsistencies when trying to reproduce results.
- **Environment Variability:** Differences in computing environments, such as **operating systems**, hardware, and **installed software**, can affect the reproducibility of results.



Imagine I want to share a Python script with all of you...

Which version of Python is available on your laptop ?

1. Open a Terminal console
2. Type : `python --version` or `python3 --version`

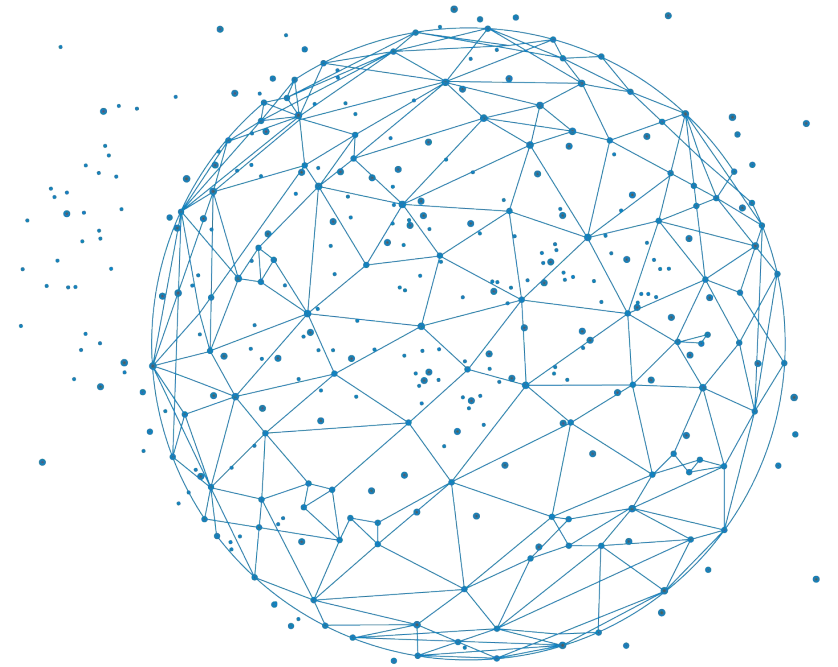




Main changes introduced in latest Python 3 versions

Version	Main Changes
3.5	Asyncio introduced for asynchronous programming, Type hints, and the @ operator for matrix multiplication.
3.6	Formatted string literals (f-strings), Underscores in numeric literals, Asynchronous generators and comprehensions.
3.7	Data classes, Context variables, Guaranteed dictionary ordering, and Performance improvements.
3.8	Assignment expressions (walrus operator :=), Positional-only parameters, and Improved type hints.
3.9	Dictionary union operators, Type hinting generics in standard collections, and Flexible function and variable annotations.
3.10	Structural pattern matching, Parenthesized context managers, and Precise error messages.
3.11	Faster runtime due to optimizations, Improved error messages with more detail, and Introduction of the ExceptionGroup exception type.
3.12	More flexible f-strings, Improved error messages, and Typing and performance enhancements.

Installing software is hard



Installing software can be a tricky operation:

Configure

Compile

Install

README License

Building Samtools

See [INSTALL](#) for complete details. [Release tarballs](#) contain generated files that have not been committed to this repository, so building the code from a Git repository requires extra steps:

```
autoheader          # Build config.h.in (this may generate a warning about
                    # AC_CONFIG_SUBDIRS - please ignore it).
autoconf -Wno-syntax # Generate the configure script
./configure          # Needed for choosing optional functionality
make
make install
```

By default, this will build against an HTSlib source tree in `../htslib`. You can alter this to a source tree elsewhere or to a previously-installed HTSlib by configuring with `--with-htslib=DIR`.

<https://github.com/samtools/samtools>

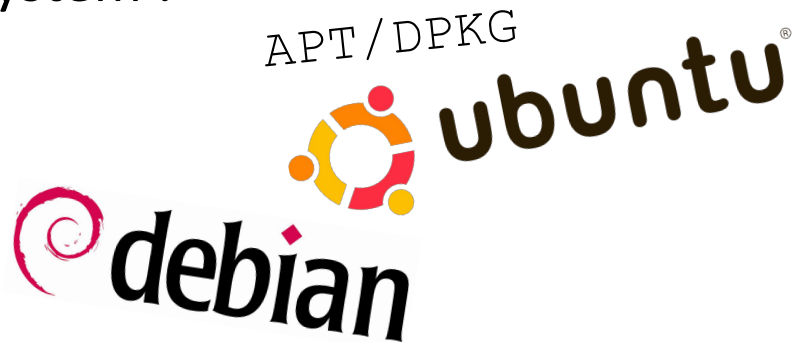
To facilitate software deployment, a number of package systems are available:

- offers pre-compiled software versions
- automatically manages dependencies
- offers updates

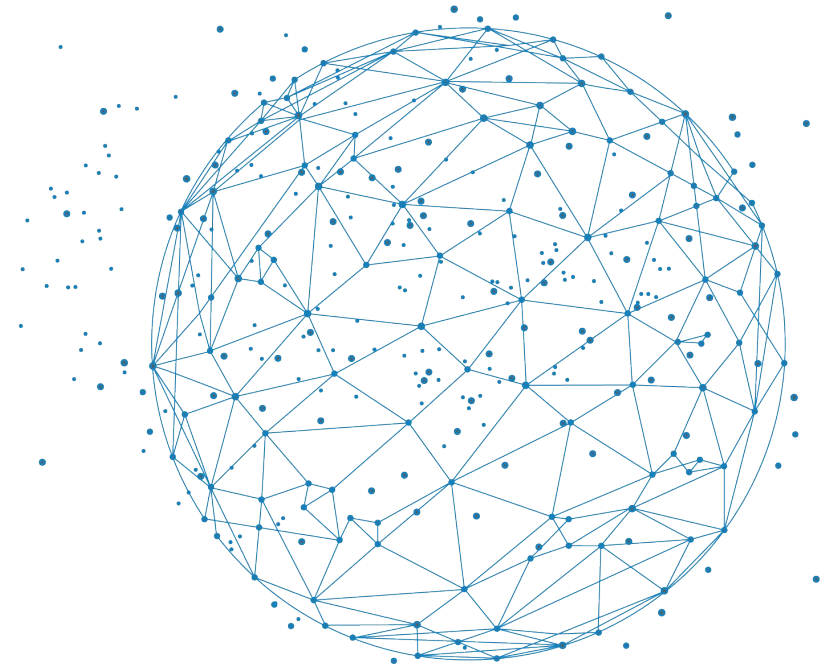
However, these package systems have a number of **drawbacks**:

- require administrative rights
- often only one version of a tool can be installed (sometimes without choice)
- poor support of scientific software
- hard to contribute

Most Linux distributions include a package system :



Introduction to CONDA



What is Conda?

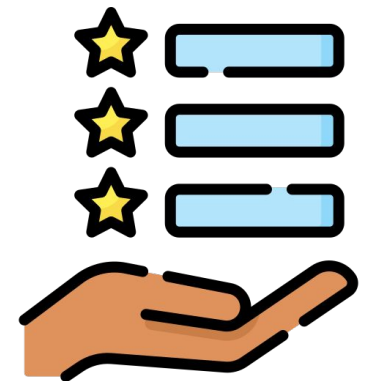
Conda is an open-source package management system and environment management system.

It helps in installing, running, and updating tools and their dependencies.



Key Features

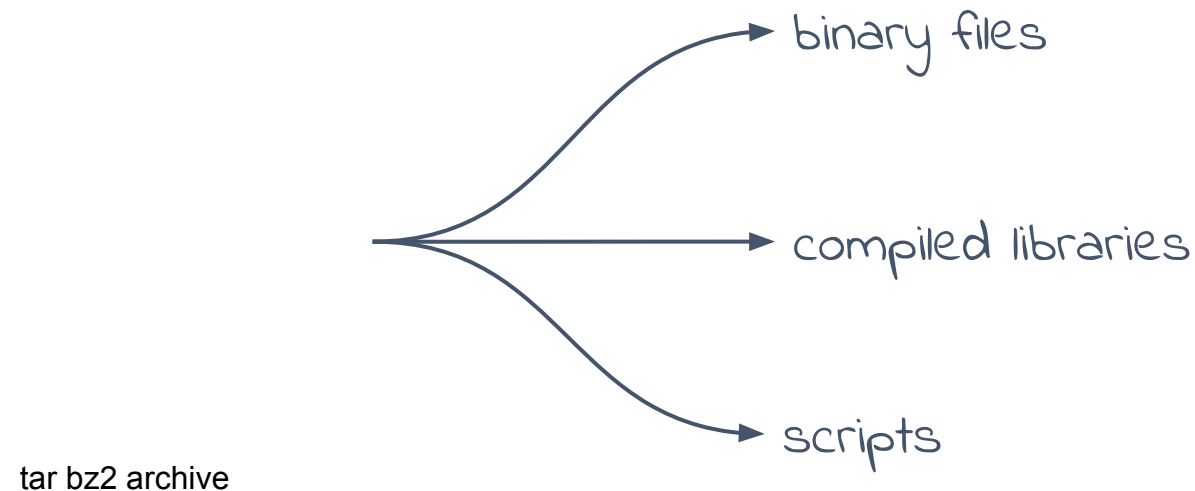
- Does **not** require administrator rights
- Allows you to **choose the version** of the tool you want to install
- Allows **multiple versions of the same tool** to coexist (through isolated environments)
- Proposes many tools in the **scientific field** (bioconda)
- A very **open contribution** model
- Compatible with any **Linux** distribution
- Also compatible with **MacOS** and **Windows** (in theory)





How does it work ?

- Each software and library is available in the form of a package (a tar bz2 archive).
- A package contains the compiled version of the software and the list of packages on which it depends.

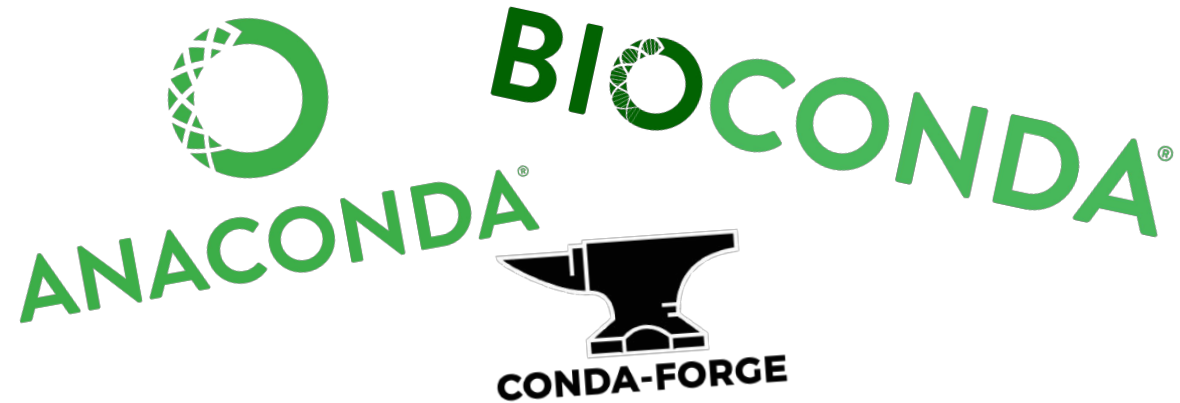


How does it work ?

- The packages are hosted on a central server: **anaconda.org**



Free



Organised by channels



By the way, what is a channel ?

A channel is a set of packages available on the conda repositories and each channel is managed by a dedicated organisation.



The creators of conda *offer* a set of packages in the **ANACONDA** channel.

For several years, two main channels propose most of the packages:

- **CONDA-FORGE** : for all user tools, languages and libraries
- **BIOCONDA** : for bioinformatics tools and libraries.

There are also numerous channels proposed by conda users. On these channels you can sometimes find tools in more recent versions, **but reliability is not guaranteed.**

How does it work ?

- The anaconda.org website has a search engine that lets you look for packages across all the channels.

The screenshot shows the Anaconda.org website with a search bar containing 'samtools'. Below the search bar, there are filters for Type, Access, and Platform, all set to 'All'. The search results are displayed in a table with columns for Favorites, Downloads, Artifact (owner / artifact), and Platforms. The table lists several 'samtools' packages from different channels, including bioconda, soil, and BioBuilds.

⬆ Favorites	⬆ Downloads	⬆ Artifact (owner / artifact)	Platforms
25	4977181	bioconda / samtools 1.19.2 Tools for dealing with SAM, BAM and CRAM files	linux-64 linux-aarch64 osx-64
2	878734	bioconda / bioconductor-rsamtools 2.18.0 Binary alignment (BAM), FASTA, variant call (BCF), and tabix file import	linux-64 osx-64
3	215435	soil / samtools 1.6 Tools for dealing with SAM, BAM and CRAM files	linux-64 osx-64
0	193215	bioconda / perl-bio-samtools 1.43 Read SAM/BAM files	linux-64 osx-64
0	8097	bioconda / msamtools 1.1.3 microbiome-related extension to samtools	linux-64 osx-64
0	5555	BioBuilds / samtools 1.6.0	linux-64 linux-

How does it work ?

- For a given software version, there are several archives corresponding to the different OS and CPU architectures for which the software is available.

bioconda / packages / samtools ☆ 25

Tools for dealing with SAM, BAM and CRAM files

Conda **Files** Labels Badges

Filters

Type: All ▾ Version: 1.19.1 ▾ Label: All ▾

☐	Type	Size	Name	Uploaded	Downloads	Labels
☐	conda	465.5 kB	osx-64/samtools-1.19.1-hd510865_0.tar.bz2	1 month and 22 days ago	164	main
☐	conda	462.2 kB	linux-64/samtools-1.19.1-h50ea8bc_0.tar.bz2	1 month and 22 days ago	1018	main

Conda environments

With conda, **each piece of software must be installed in an environment.**

An environment is a folder containing all the files needed for the software to work.

This folder looks like **a miniature operating system.**

You can install multiple software in an environment but **only one version of a software in a given environment.**

You can create **as many environments as you want** each containing their own set of software.



Creating a conda environment :

```
$ conda create -n my_env  
Empty environment created at prefix: ~/.conda/envs/my_env
```

Activate a Conda environment :

To choose which conda environment you want to use, you have to **activate** it

```
$ conda activate my_env  
(my_env) $
```



Installing software

```
(my_environment) $ conda install -c bioconda fastqc=0.12.1  
...
```



Specifying a channel is optional

```
(pipeline)$ conda install -c bioconda fastqc=0.12.1
```

```
...
```

```
Updating specs:
```

```
- fastqc=0.12.1
```

Package	Version	Build	Channel	Size
---------	---------	-------	---------	------

Install:

+ _libgcc_mutex	0.1	conda_forge	conda-forge	Cached
+ _openmp_mutex	4.5	2_gnu	conda-forge	Cached
+ fastqc	0.12.1	hdfd78af_0	bioconda	12MB
+ font-ttf-dejavu-sans-mono	2.37	hab24e00_0	conda-forge	Cached
+ fontconfig	2.15.0	h7e30c49_1	conda-forge	Cached
+ freetype	2.13.3	ha770c72_1	conda-forge	Cached
+ libexpat	2.7.0	h5888daf_0	conda-forge	Cached
+ libfreetype	2.13.3	ha770c72_1	conda-forge	Cached
+ libfreetype6	2.13.3	h48d6fc4_1	conda-forge	Cached
+ libgcc	15.1.0	h767d61c_2	conda-forge	Cached
+ libgcc-ng	15.1.0	h69a702a_2	conda-forge	Cached
+ libgomp	15.1.0	h767d61c_2	conda-forge	Cached
+ libpng	1.6.47	h943b412_0	conda-forge	Cached
+ libuuid	2.38.1	h0b41bf4_0	conda-forge	Cached
+ libxcrypt	4.4.36	hd590300_1	conda-forge	Cached
+ libzlib	1.3.1	hb9d3cd8_2	conda-forge	Cached
+ openjdk	11.0.1	h516909a_1016	conda-forge	184MB
+ perl	5.32.1	7_hd590300_perl5	conda-forge	13MB
+ zlib	1.3.1	hb9d3cd8_2	conda-forge	Cached

Summary:

Install: 19 packages

Total download: 209MB

```
Confirm changes: [Y/n]
```

```
...
```

Conda proposes a large list of packages to install.

These are all dependencies required to run **fastqc**.

For each package it indicates the source channel and the version it will deploy.

We can see that it has found **fastqc** in the **bioconda** channel but that most of the dependencies come from the **conda-forge** channel.

The **bioconda** channel offers tools specifically for bioinformatics.

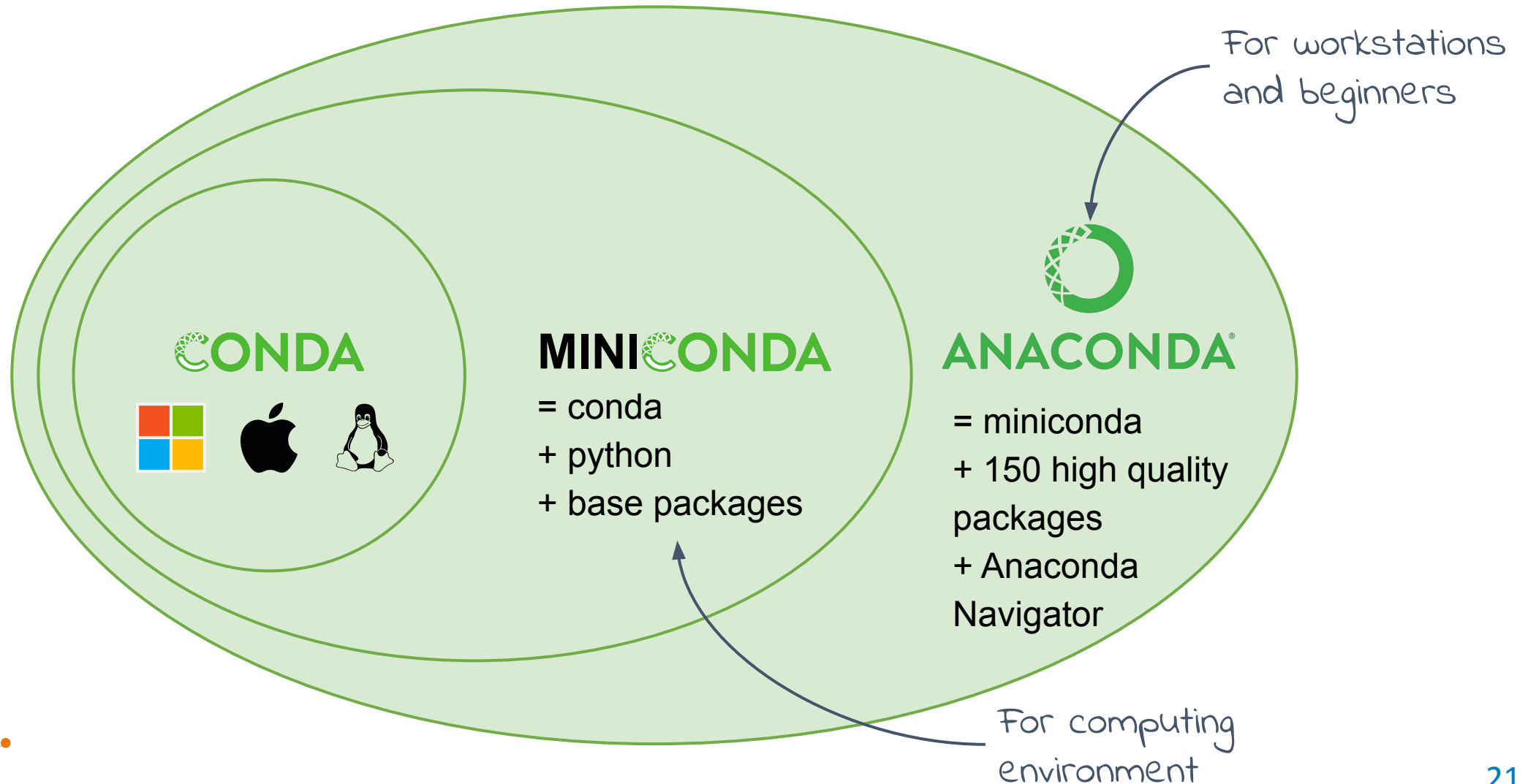
The **conda-forge** channel offers genetic tools and libraries.



Benefits for Reproducibility

- Software catalog: Anaconda offers a vast catalog of software and maintains all known versions for each software package.
- Isolation: Environments can be isolated to avoid conflicts between package versions.
- Sharing Environments: Environments can be shared using environment files (environment.yml), ensuring that others can replicate the exact setup.

Conda is available in different flavors



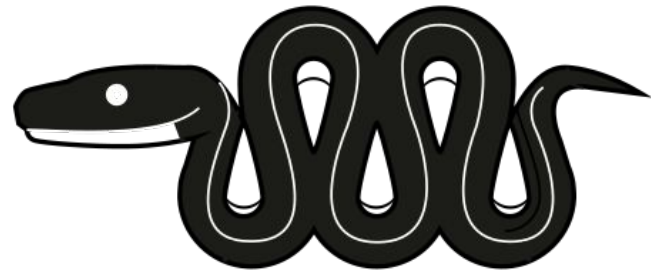
A faster solver as been developed for conda : libmamba

*This library has been developed by a french company called **QuantStack***

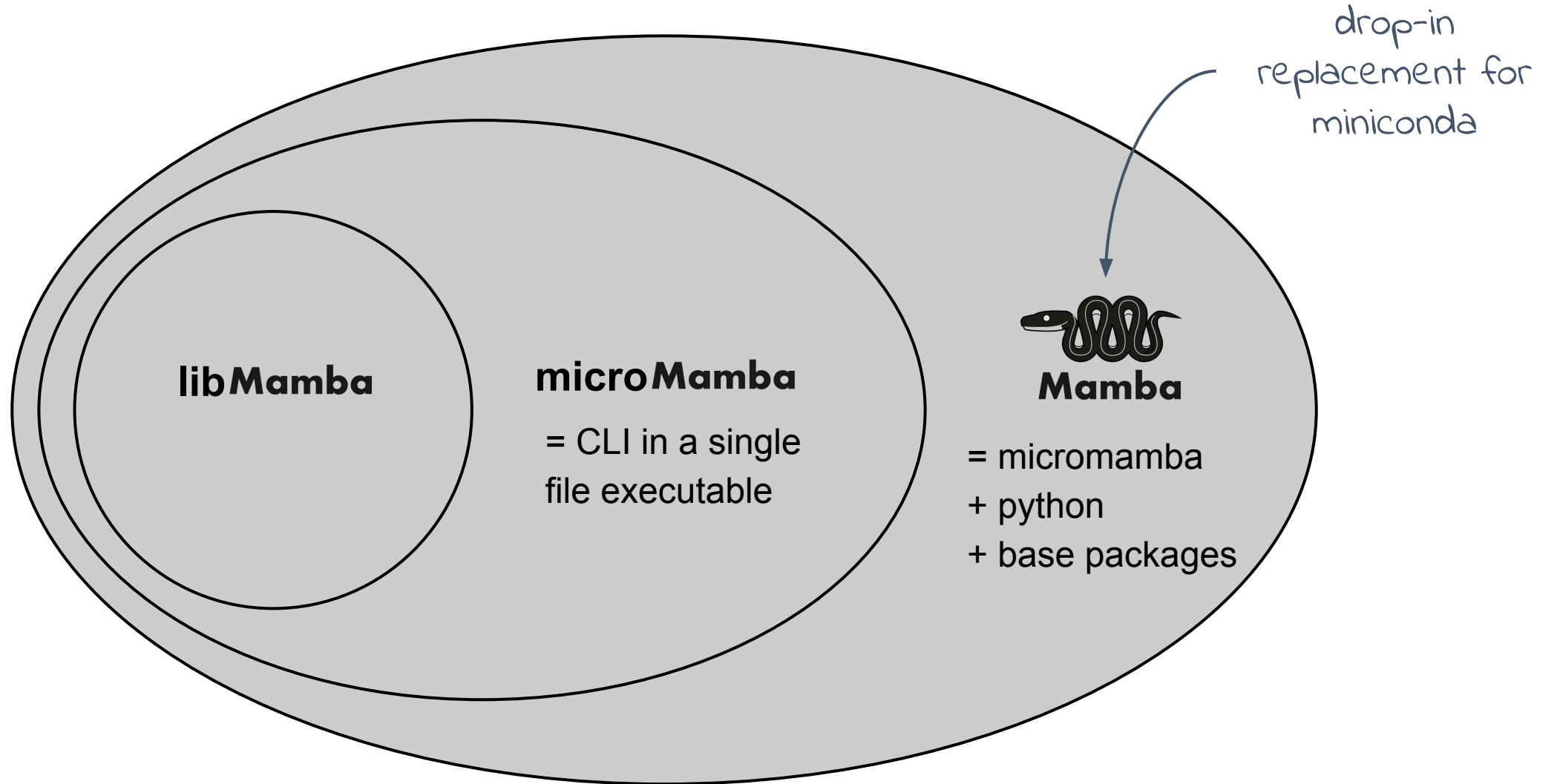


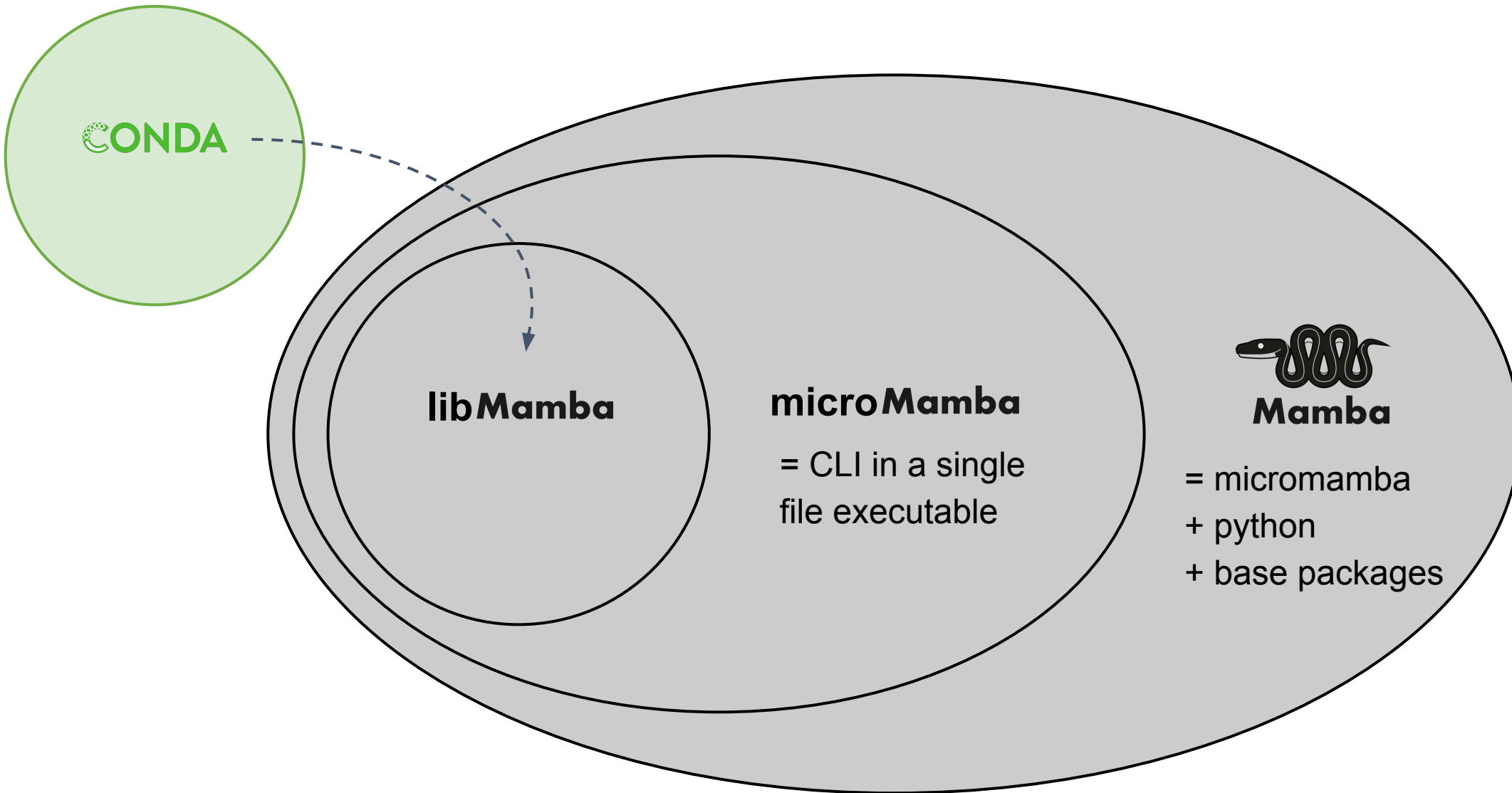
A new package manager based on libmamba is now available : mamba

It uses the same channels as conda.



Mamba





conda or mamba run natively on many operating systems without the need to install special system packages. **They can therefore be used as a non-administrator user.**

Similarly, the same installation of conda or mamba can run on several OS. This is the case on the Core cluster where some of the compute nodes are running Centos and others Ubuntu.

conda or mamba also support OS updates very well.



For the most light and simple experience, we recommend using **micromamba**

It can be installed with one single command :

```
$ "${SHELL}" <(curl -L micro.mamba.pm/install.sh)
```

After installation, the **micromamba** command will be available on your computer.

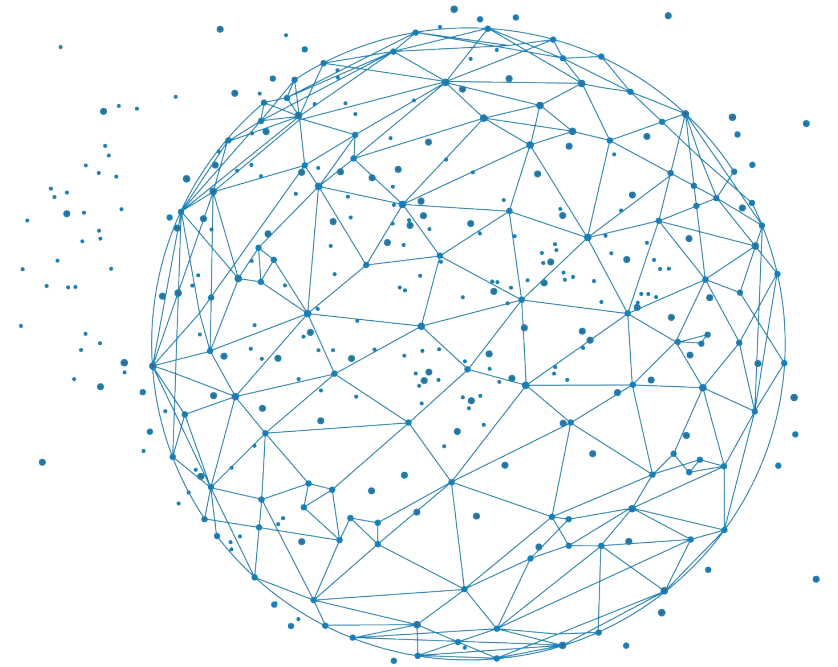
Same tool, different names :

If you rather install miniconda or anaconda, the command will be **conda**

If you choose to install mamba, the command will be **mamba**

micromamba, **mamba** and **conda** proposes all the same command line interface.

A few words about encapsulation



Let's say we want to install RStudio...

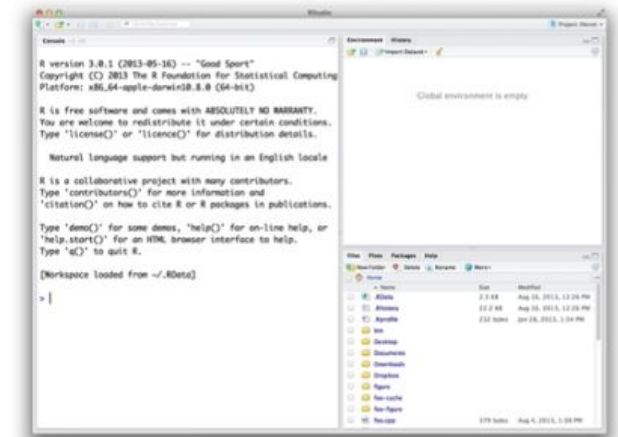
MacOS



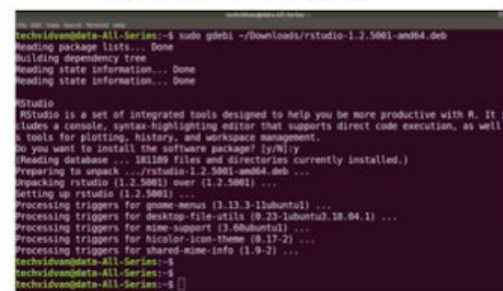
Windows



Use Rstudio

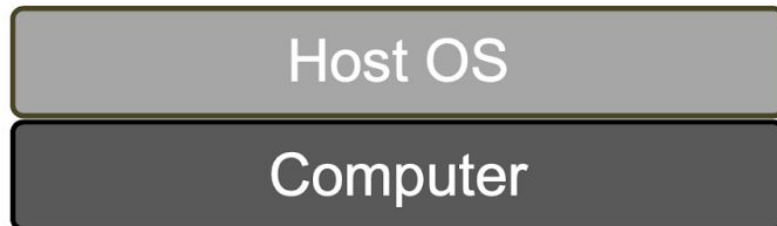


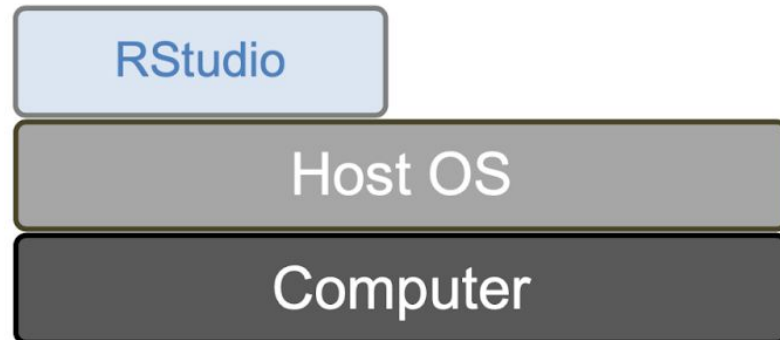
Unix-based





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



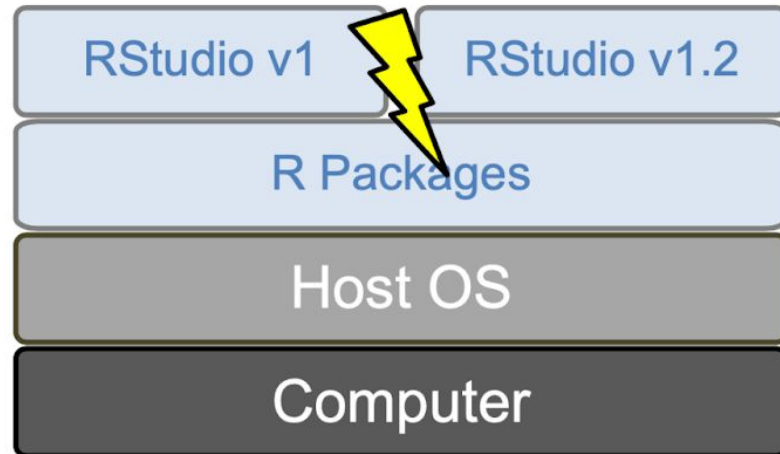


We started with a computer using a specific OS...

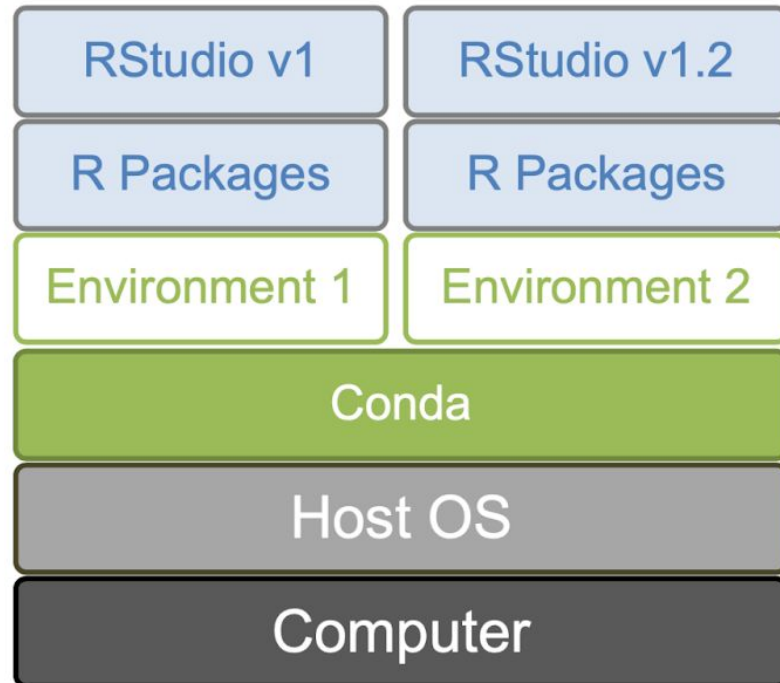
And inside this environment, we installed a new application.



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...

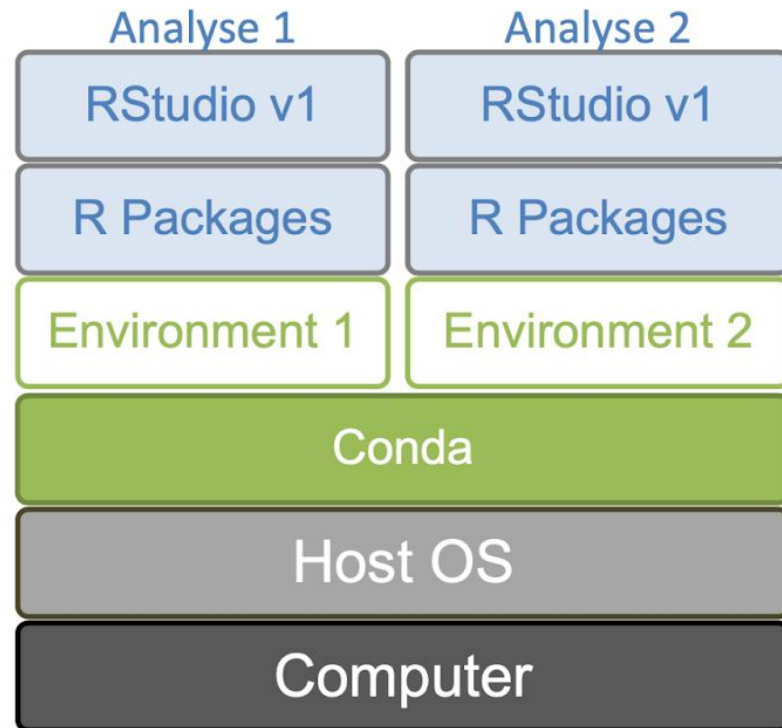


Usually dependencies of different applications don't interfere.
But what if we want to test the latest version of our favourite tool?
There might be conflicts...



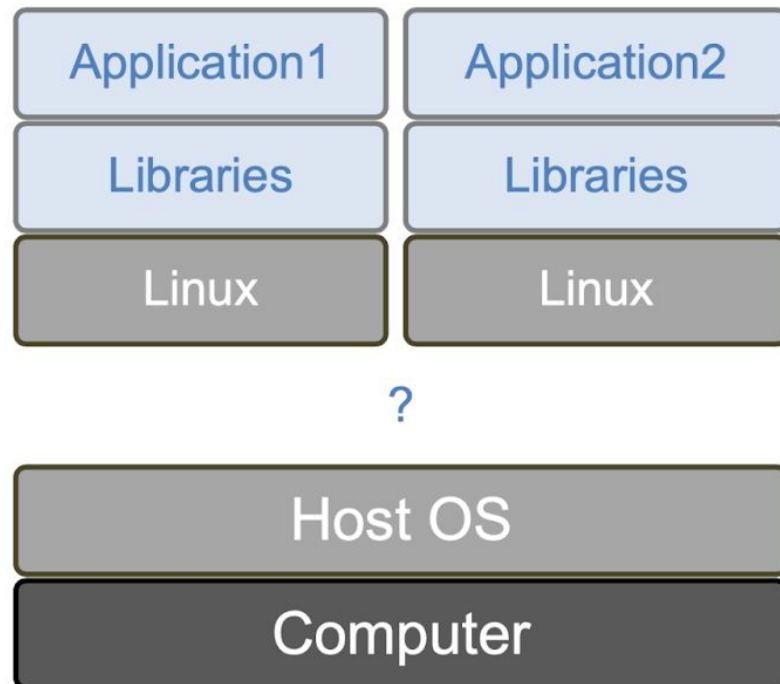
Idea : create separated environments for each application.

CONDA

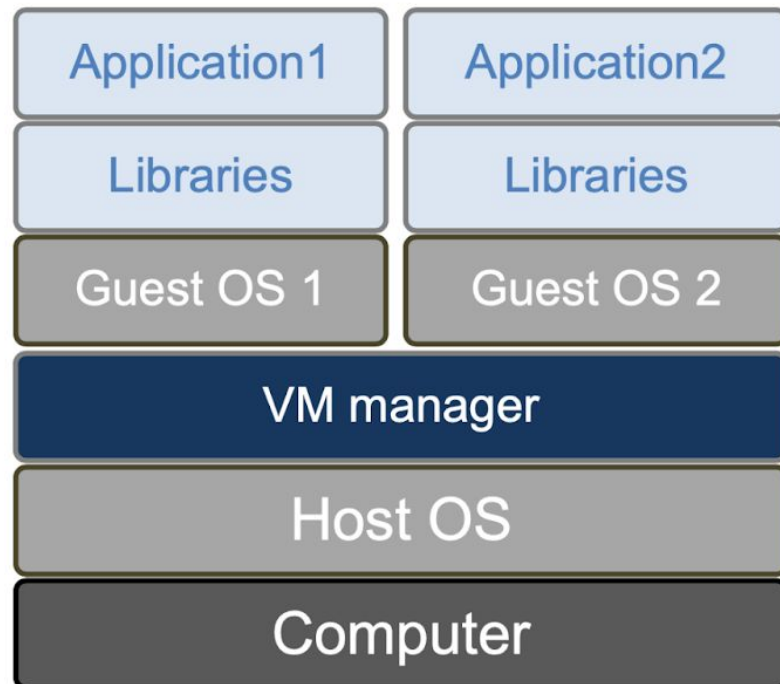


Idea : create separated environments for each application.

More versatile: create a new environment per analysis.



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

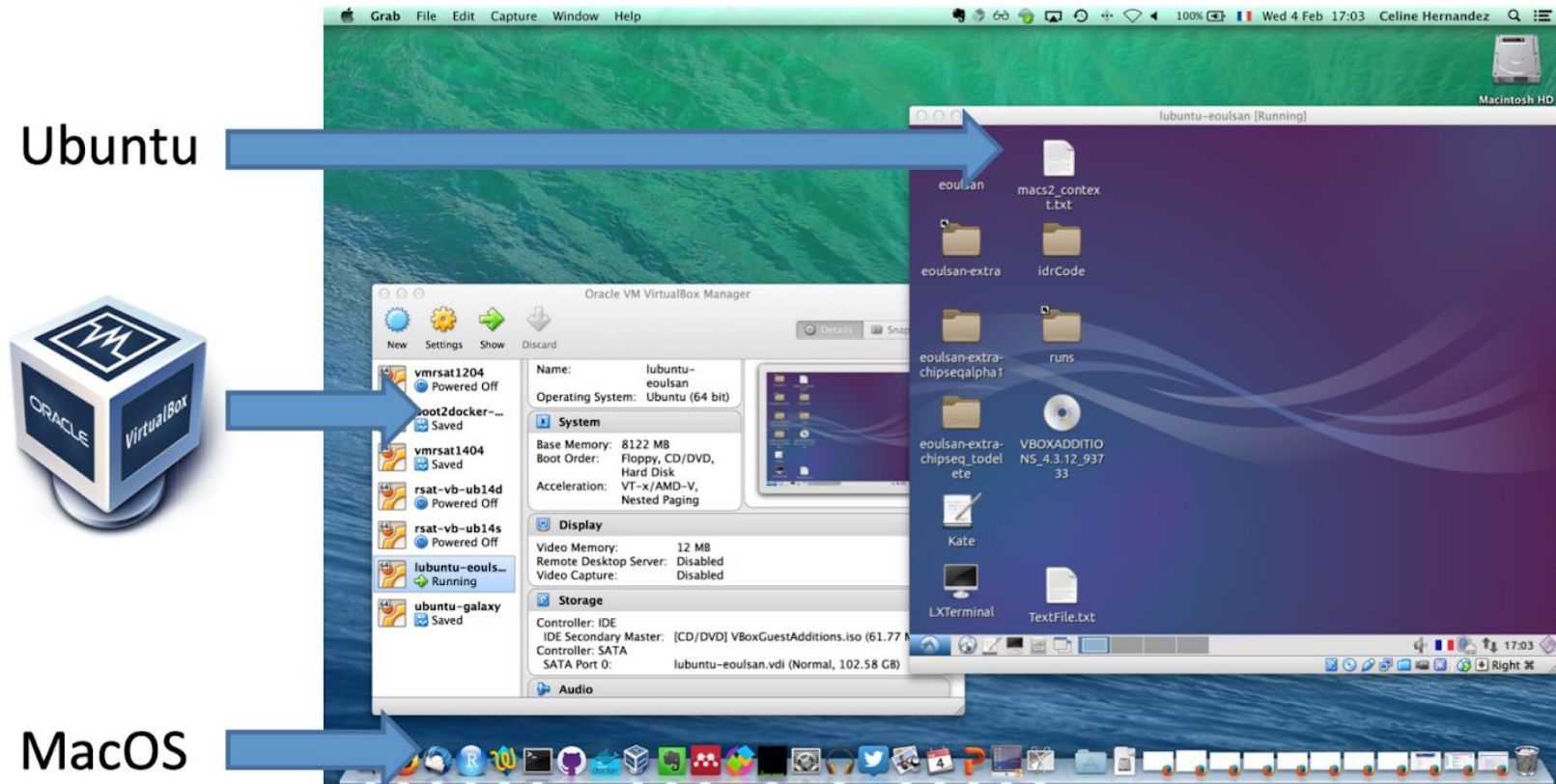


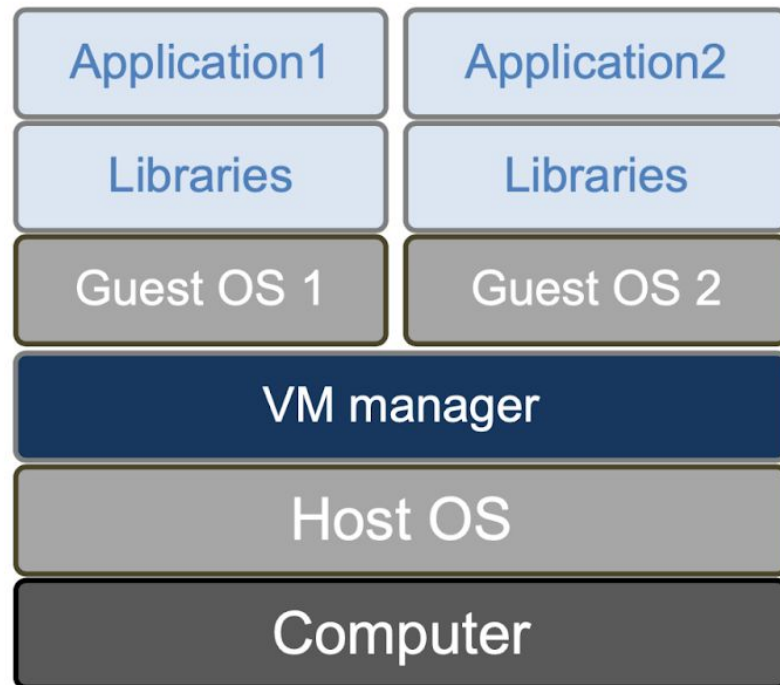
Idea: use virtual machines

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



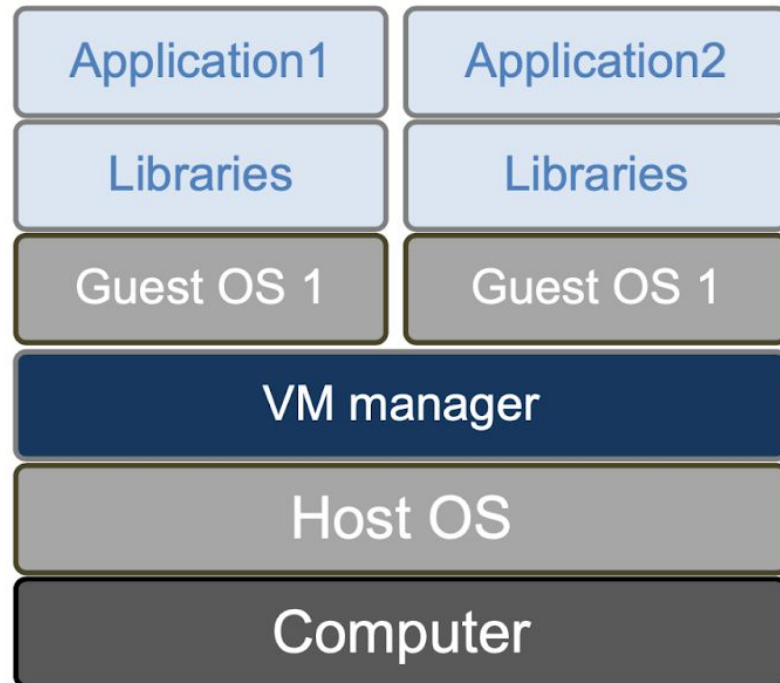


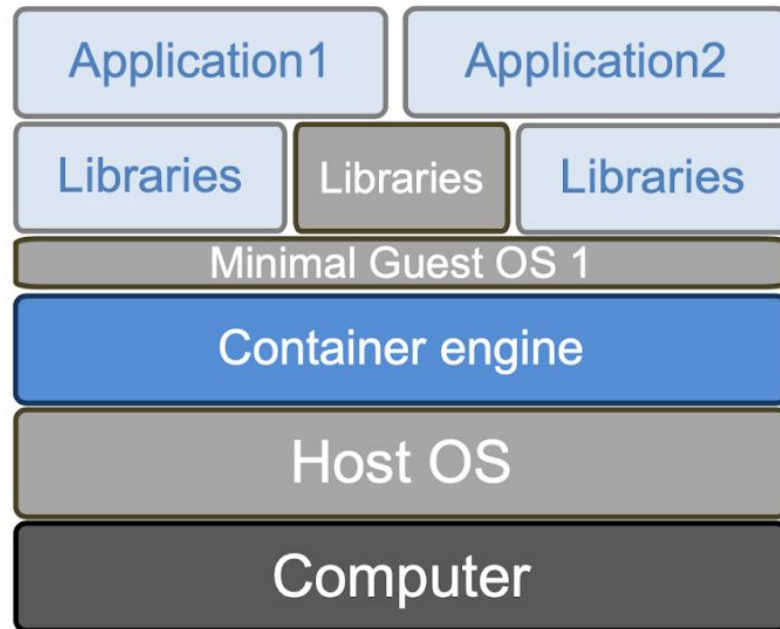
Idea: use virtual machines

Pros: transferable independent environments

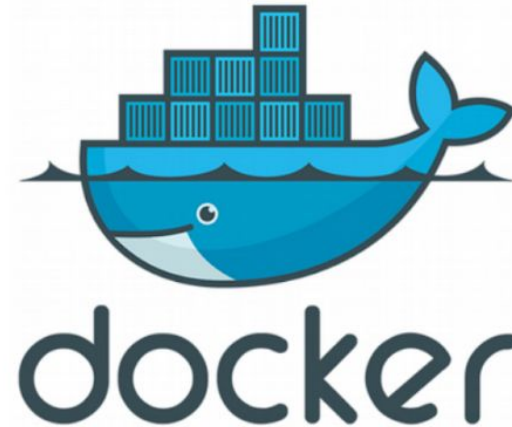
Cons:

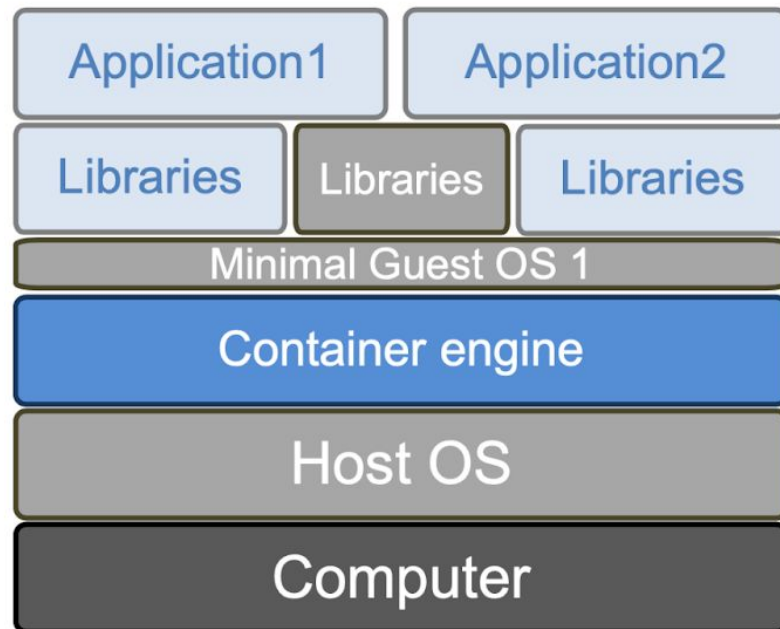
- Redundancy between VMs
- Heavy to set up
- No automation





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

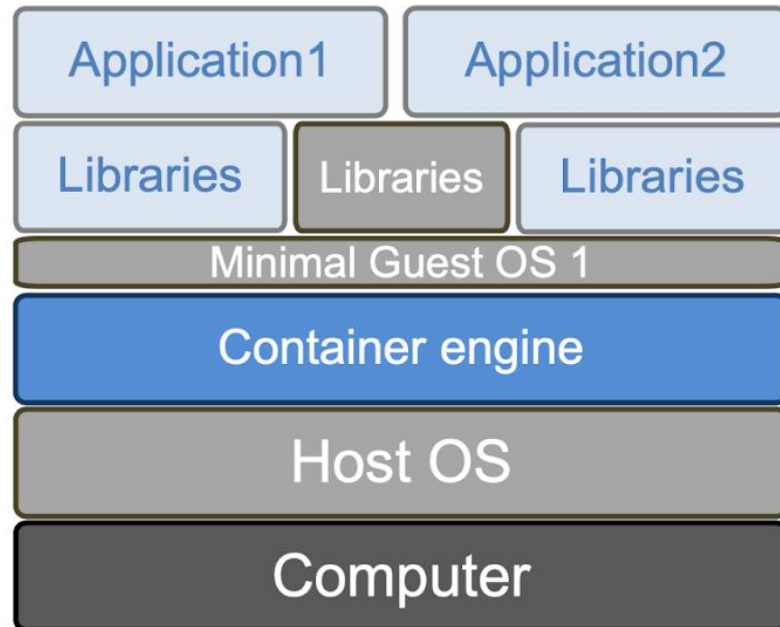




OS virtualisation vs hardware virtualisation

Pros:

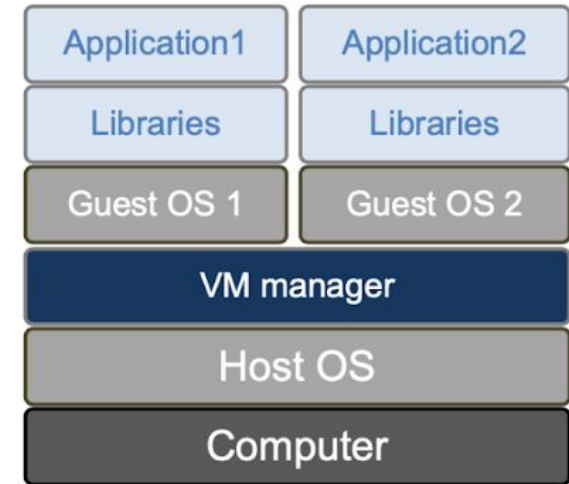
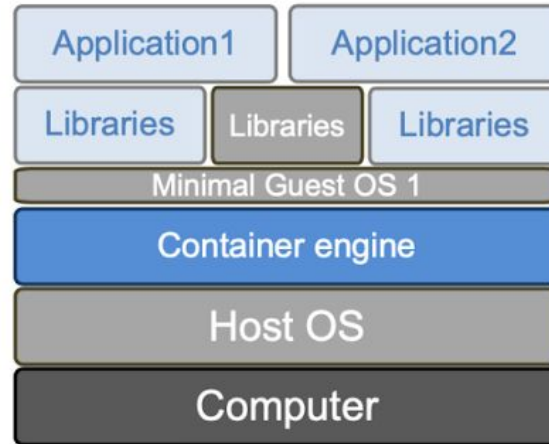
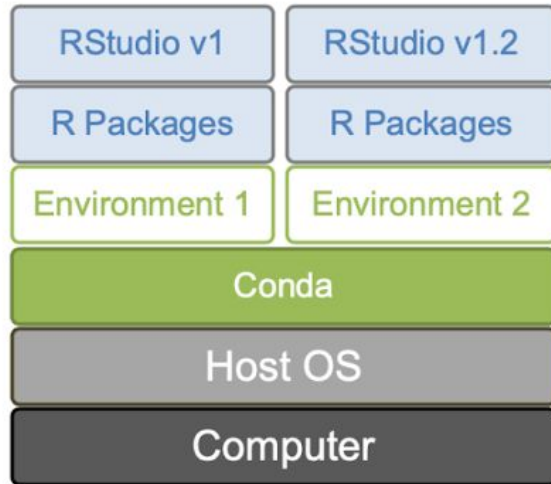
- Speed
 - ▶ Installation is faster
 - ▶ No boot time
- Lightweight
 - ▶ Minimal base OS
 - ▶ Minimal libraries and application set
- Easy sharing of applications



Cons:

- Singularity to use images on a cluster

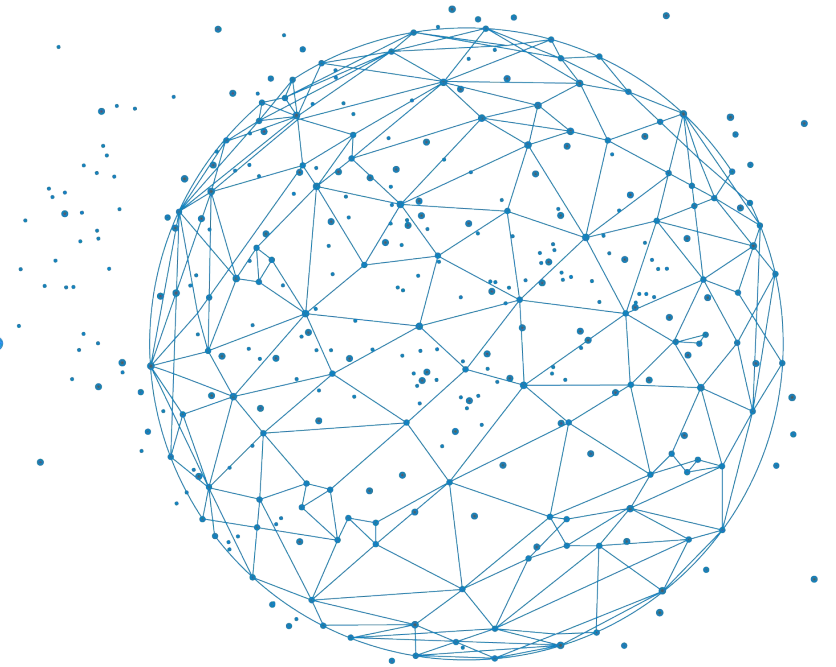
Encapsulation



CONDA



Introduction to docker



What is Docker?

Docker is a platform designed to help developers build, share, and run container applications.

Containers allow a developer to package up an application with all the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

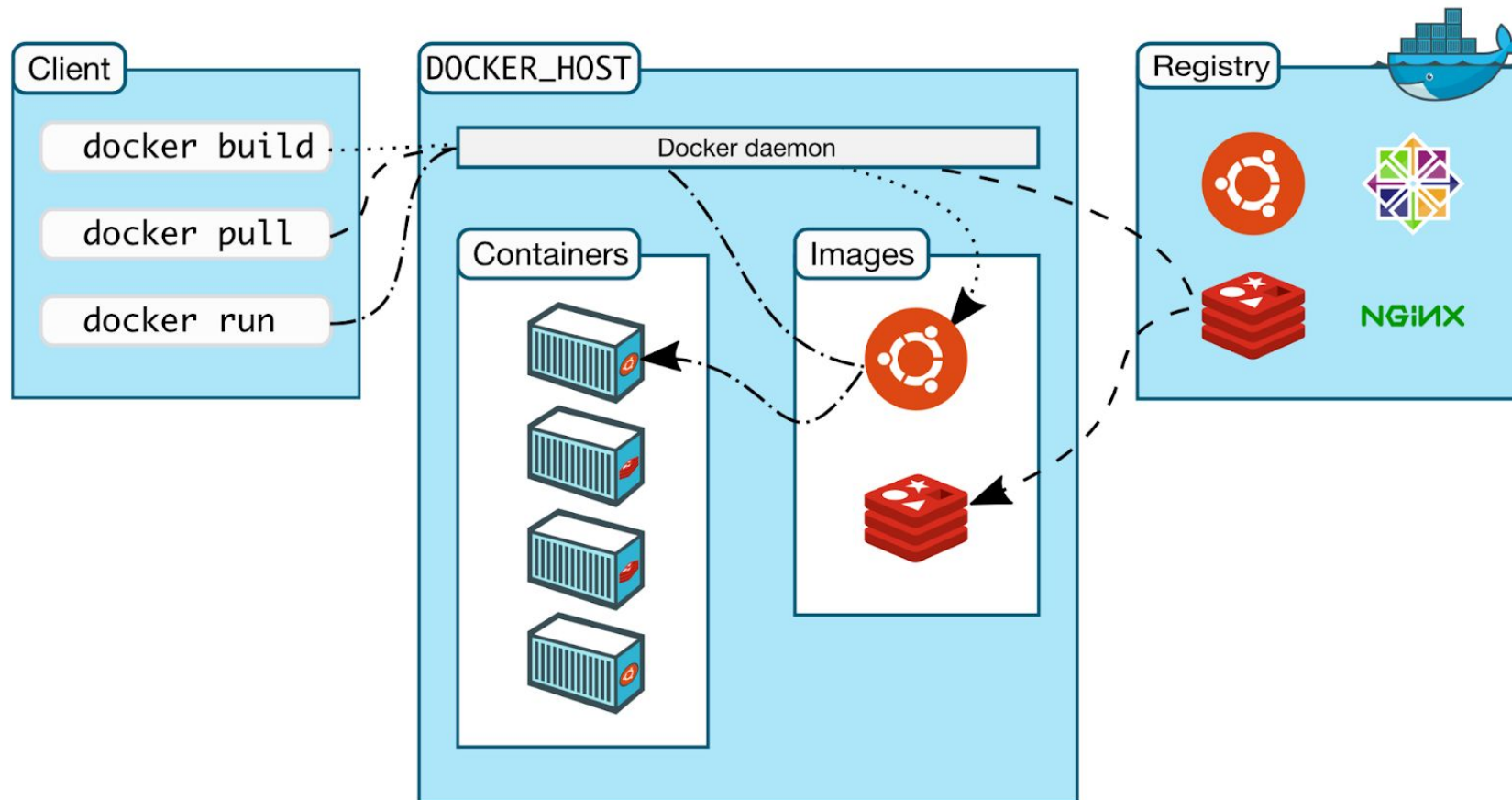
It is often defined as lightweight virtualization.



Key Features

- Docker uses **containerization** to ensure that applications run consistently across different computing environments.
- Pre-built **images** can be used to quickly set up environments.
- **Registries** are available for sharing and accessing Docker images. The most popular is the Docker Hub.

How does it work ?



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



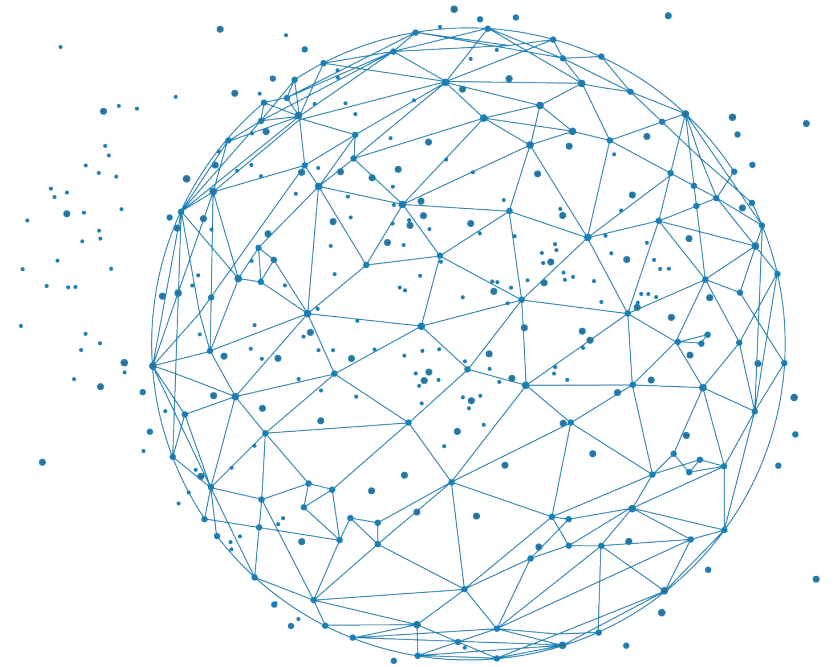
Benefits for Reproducibility

- Consistency: Ensures that the application runs the same way regardless of where it is deployed.
- Isolation: Containers provide process and filesystem isolation, which helps in avoiding conflicts.
- Version Control: Docker images can be versioned and shared, making it easy to replicate environments.

Introduction to



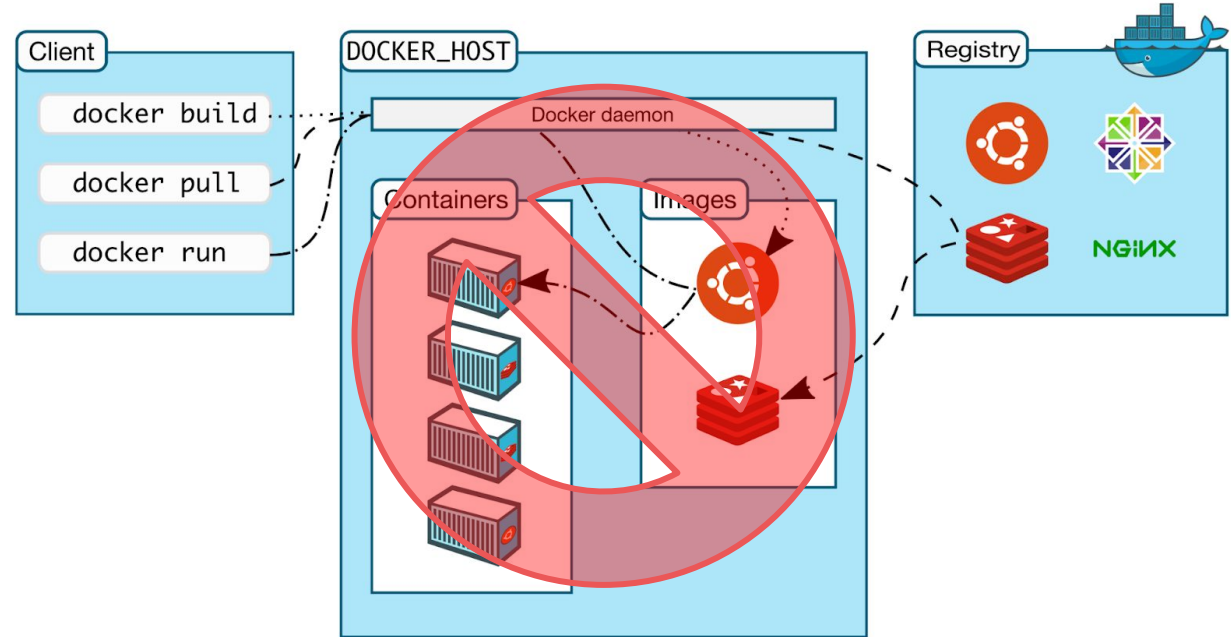
APPTAINER



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 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**



Key Features

- Optimized for use in HPC environments where Docker might not be suitable.
- Designed with security in mind, allowing users to run containers without requiring root privileges.
- Can run Docker images, making it easy to transition from Docker.



Running a Docker image on an HPC cluster with Apptainer

```
# load apptainer
$ module load apptainer
```



Running a Docker image on an HPC cluster with Apptainer

```
$ apptainer build cowsay.sif docker://rancher/cowsay
INFO:      Starting build...
Copying blob 34d5e986f175 done |
Copying blob dd05e66d8cea done |
Copying blob cbdbe7a5bc2a done |
Copying blob 13eefd6dfff68 done |
Copying config 223a921ebc done |
Writing manifest to image destination
2025/06/05 22:39:16 info unpack layer: sha256:cbdbe7a5bc2a134ca8ec91be58565ec07d037386d1f1d8385
2025/06/05 22:39:16 info unpack layer: sha256:dd05e66d8ceaaf3d7cf6712075c8c28099aa003d0c625be8b
2025/06/05 22:39:17 info unpack layer: sha256:34d5e986f1757ac7e1b094d89b7ba9e40ee5b614fc91ad2a9
2025/06/05 22:39:17 info unpack layer: sha256:13eefd6dfff6843c9689bb227b3eeb54c1f427c1397e029866
INFO:      Creating SIF file...
INFO:      Build complete: cowsay.sif
```



Running a Docker image on an HPC cluster with Apptainer

```
$ apptainer run cowsay.sif "Hello FAIRBioinfo"
```

```
< Hello FAIRBioinfo >
```

```
-----
```

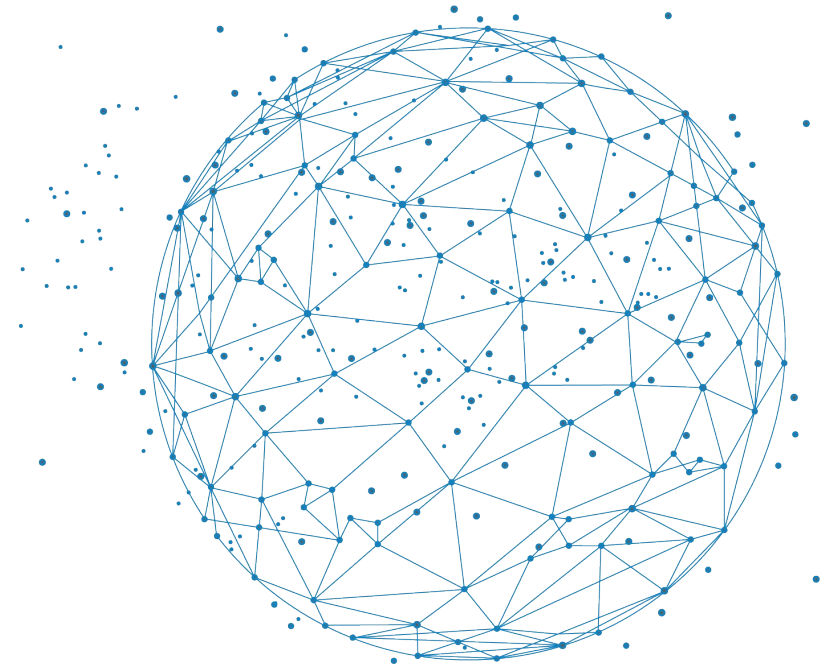
```
  \      ^__^
   \    (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||
```



Benefits for Reproducibility

- Portability: Containers can be easily moved and run on different HPC systems.
- Reproducibility: Ensures that the computational environment is consistent and can be replicated.
- Security: Enhanced security features make it suitable for multi-user HPC environments.

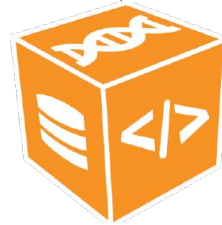
A brief summary





Tool	Primary Use Case	Key Strengths	Potential Weaknesses
Conda	Package and environment management	Easy to use, cross-platform, no root access required	Limited to package built by the community, less isolation compared to containers
Docker	Containerization	Highly portable, strong isolation, extensive image repository	Requires root access, not ideal for HPC environments
Apptainer	Containerization for HPC	Secure, no root access required, compatible with Docker images	Less user-friendly for non-HPC use cases

BIOCONDA®



BIOCONTAINER





Thank you for your
attention !