

Environment manager from your OS to your environment

Encapsulation levels using docker, conda¹



Pierre Marin
pierre.marin@uca.fr

Université Clermont Auvergne, AuBi, Mésocentre

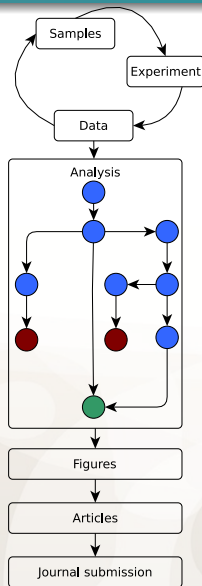
3 octobre 2022



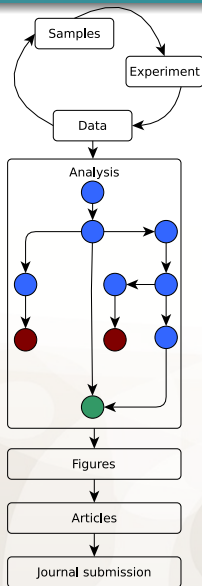
1. This work is derived from the IFB and I2BC team members

- 1 A common use-case
 - Retry my results
 - The use of packaging
 - Example with R
- 2 Manage your local environment
 - How conda works
- 3 Manage your hardware configuration
 - How virtual manager works
- 4 Manage your OS configuration
 - How container works
- 5 Conda ecosystem
 - a case of bioconda

A classic use case



A classic use case



Your manuscript as submitted

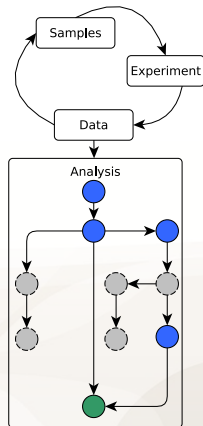
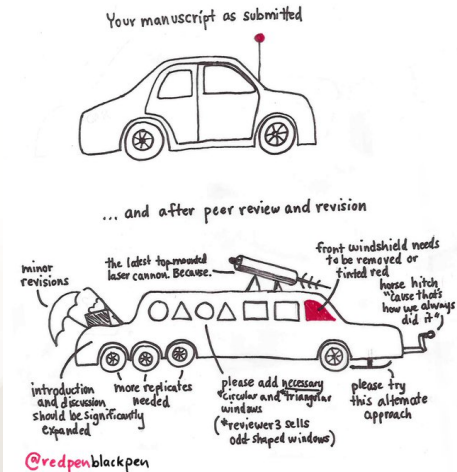
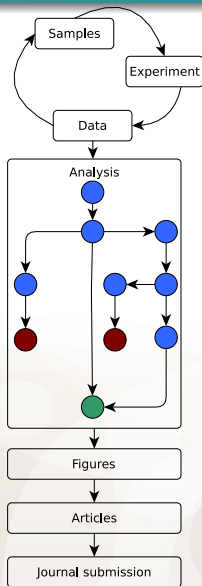


... and after peer review and revision



@redpenblackpen

A classic use case



A classic use case

What are the changes ?

- Tool version
- Packages
- Environment variables
- OS version
- The computer
- ...

■ Tool compatibility troubles

- Python version ? 2.7, 3.8...
- Which tool version ?
- Installation without root access
- coexistence bewteen severals versions, libraries

Encapsulation levels

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

- Hardware virtualisation (virtual machines)



Encapsulation levels

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

- Hardware virtualisation (virtual machines)



- OS virtualisation (images and containers)



Encapsulation levels

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

- Hardware virtualisation (virtual machines)



- OS virtualisation (images and containers)



- Environment management (package manager) **CONDA**

Example of R and package installation

Classical installation


- Start with a computer and a specific OS

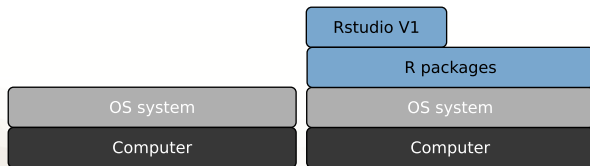
OS system

Computer

Example of R and package installation



Classical installation

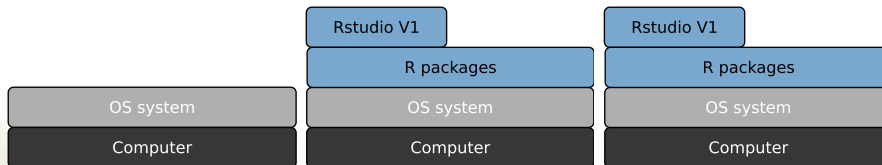
- Start with a computer and a specific OS
- Inside, we installed a new  application



Example of R and package installation




Classical installation

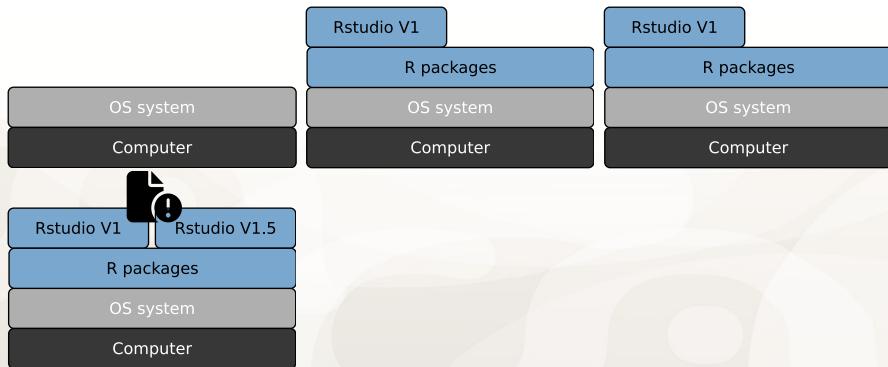
- Start with a computer and a specific OS
- Inside, we installed a new  application
-  need some dependencies



Example of R and package installation

Classical installation

- Start with a computer and a specific OS
- Inside, we installed a new  application
-  need some dependencies
- we tested the last  version -> might be conflicts



Some recommendations²

- A package first

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible
- Keep data outside of the container

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible
- Keep data outside of the container
- Add functional testing logic

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible
- Keep data outside of the container
- Add functional testing logic
- Check the license of the software

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible
- Keep data outside of the container
- Add functional testing logic
- Check the license of the software
- Make your package or container discoverable

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible
- Keep data outside of the container
- Add functional testing logic
- Check the license of the software
- Make your package or container discoverable
- Provide reproducible and documented builds

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Some recommendations²

- A package first
- One tool, one container
- Tool and container versions should be explicit
- Avoid using ENTRYPOINT
- Reduce the size of your container as much as possible
- Keep data outside of the container
- Add functional testing logic
- Check the license of the software
- Make your package or container discoverable
- Provide reproducible and documented builds
- Provide helpful usage message

2. Recommendations for the packaging and containerizing of bioinformatics software Gruening, F1000 Research, 2019.
DOI 10.12688/f1000research.15140.2

Example of R and package installation

Conda use

- The idea is to separate each application in here own environment **CONDA**



Conda env

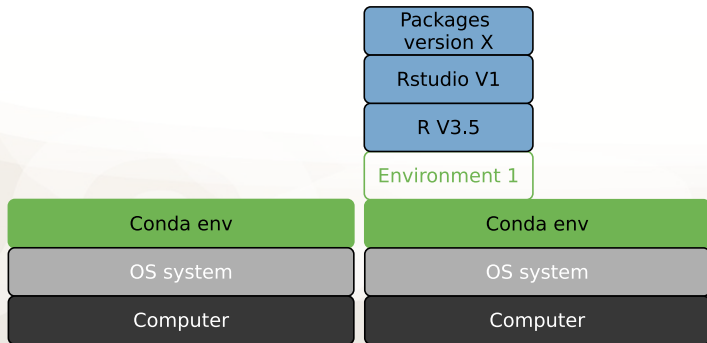
OS system

Computer

Example of R and package installation

Conda use

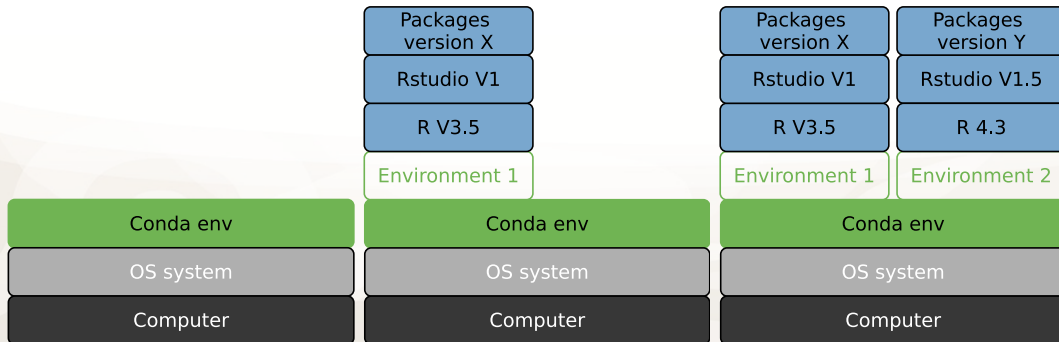
- The idea is to separate each application in here own environment **CONDA**
- A tool version, a conda environment



Example of R and package installation

Conda use

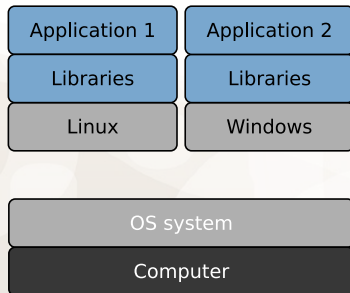
- The idea is to separate each application in here own environment **CONDA**
- A tool version, a conda environment
- Create a new environment for my new tool version, my analysis...



Example of R and package installation

hardware virtualisation

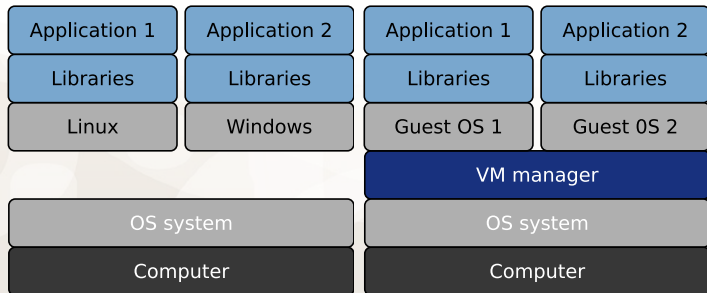
- If we want a software from a different OS ?



Example of R and package installation

hardware virtualisation

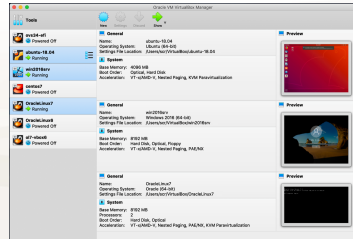
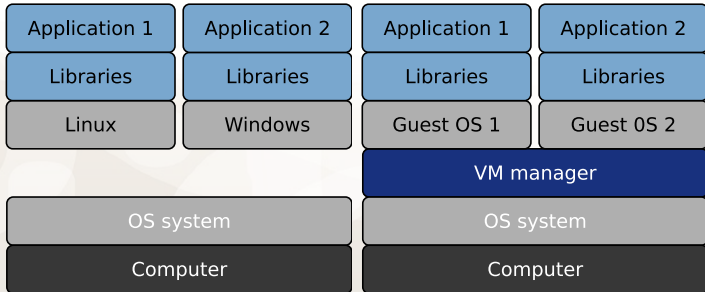
- If we want a software from a different OS ?
- Use virtual machines



Example of R and package installation

hardware virtualisation

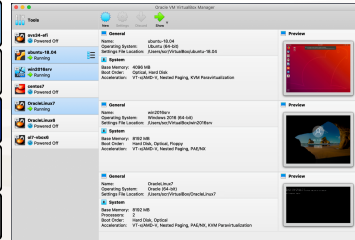
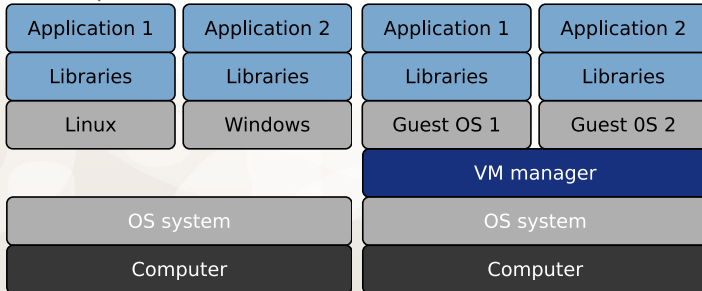
- If we want a software from a different OS ?
- Use virtual machines
- Each application get a total different and independant environment



Example of R and package installation

hardware virtualisation

- If we want a software from a different OS ?
- Use virtual machines
- Each application get a total different and independant environment
- Virtual machine could be transfered to another computer

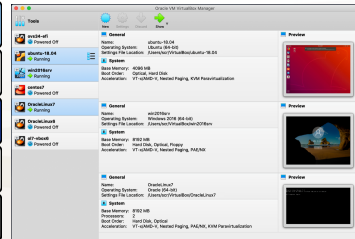
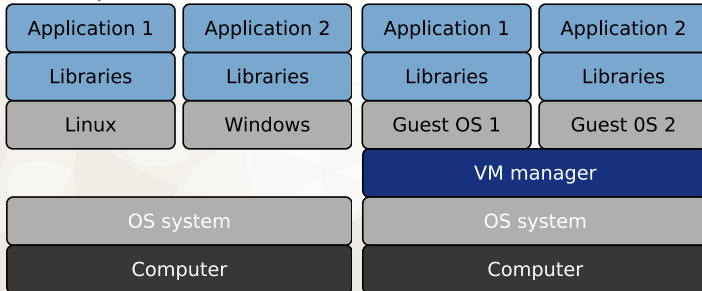


Example of R and package installation

hardware virtualisation

- If we want a software from a different OS ?
- Use virtual machines
- Each application get a total different and independant environment
- Virtual machine could be transfered to another computer

- Redundancy between VMs

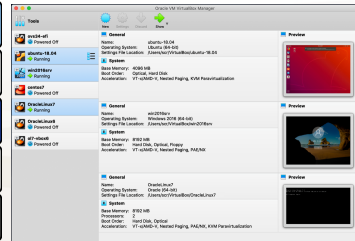
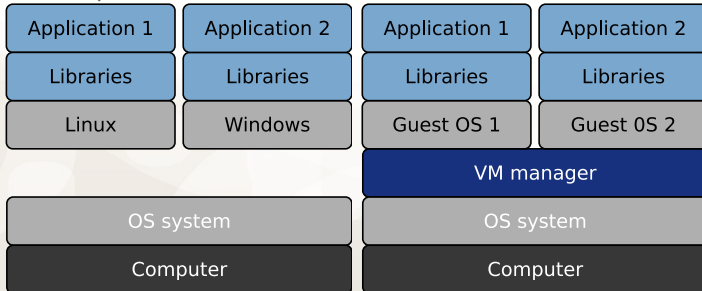


Example of R and package installation

hardware virtualisation

- If we want a software from a different OS ?
- Use virtual machines
- Each application get a total different and independant environment
- Virtual machine could be transfered to another computer

- Redundancy between VMs
- Heavy to set up

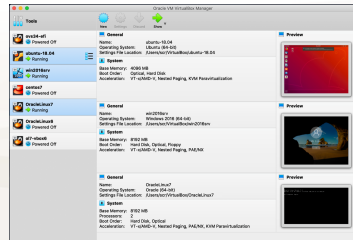
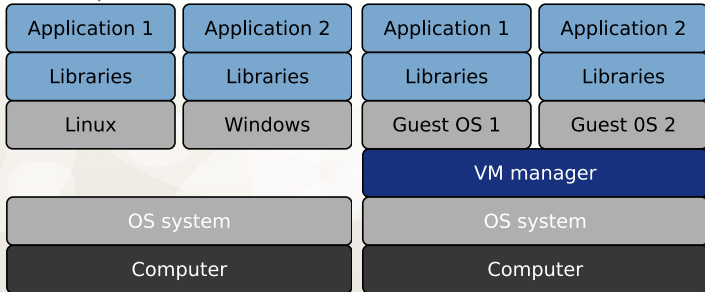


Example of R and package installation

hardware virtualisation


- If we want a software from a different OS ?
- Use virtual machines
- Each application get a total different and independant environment
- Virtual machine could be transfered to another computer

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



Example of R and package installation

OS virtualisation

- "Trick" applications into believing that they are in a different OS than the host's  docker
- Avoid redundancy
- Speed
 - Faster installation
 - No boot time
- Lightweight
 - Minimal base OS
 - Minimal set of library and global environment
 - Easy sharing of application
- No easy use on a cluster system
- Docker private company policies



3

3. <https://www.docker.com/blog/scaling-docker-to-serve-millions-more-developers-network-egress/>

Conda system

■ Anaconda

- Open source distribution
- Cross platform
- Available on cluster without admin write
- Thousands of available tool in informatic and bioinformatic

■ Miniconda

- A lightheight Anaconda version with minimal requirment
- Same advantages ad Anaconda

■ Conda

- Package manager AND environment manager
- installed with Ana pr Miniconda
- Python based but can also install tools from R, C++ or Julia...

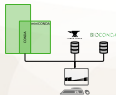


CONDA

The channels and the tools

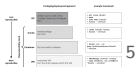
The tools are packaged and available on several "channels"

- conda-forge
- anaconda
- R
- Bioconda⁴ -> Most of the bioinformatic tools



4. Bioconda : sustainable and comprehensive software distribution for the life sciences *Grüning et al.*, Nature methods, 2018. DOI 10.1038/s41592-018-0046-7

présenter les commandes de base de conda pour lister des env lister des packages présenter le fait de gérer les versions et la compatibilité des versions présenter la résolution des environnements et la lenteur de conda donc aller vers mamba



5. Practical Computational Reproducibility in the Life Sciences Gruning et al, Cell Systems, 2018. DOI 10.1016/j.cels.2018.03.014

d