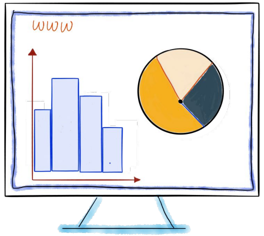
Reproducible Science in practice

tools and ideas



Arturo Sánchez Pineda (LAPP)

June 10, 2021 - ESCAPE (online) School

Some current tools **per element**

Data

Usually, labs and experiments have dedicated data repositories for their users.

Here, I want to mention to Open Access datasets repositories as examples

zenodo

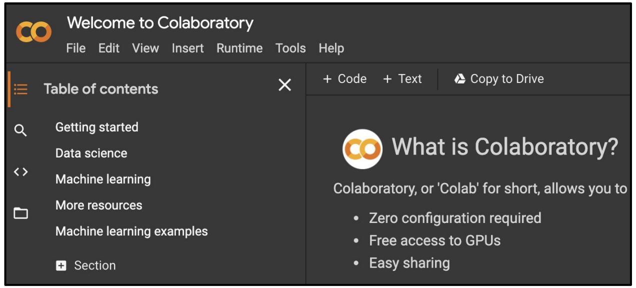
Academic Torrents

Two services that can be really useful for storage and preservation of datasets (and other digital objects)

Analysis platforms/environments like Jupyter grow in popularity and started to be some of the "standard" in several domains of data analysis

Here some examples of such tools, where Jupyter is offer as a service by public or private institutions

Analysis code SaaS


Welcome to Colaboratory

File Edit View Insert Runtime Tools Help
+ Code + Text Copy to Drive

Table of contents

- Getting started
- Data science
- Machine learning
- More resources
- Machine learning examples

Section

What is Colaboratory?

Colaboratory, or 'Colab' for short, allows you to

- Zero configuration required
- Free access to GPUs
- Easy sharing

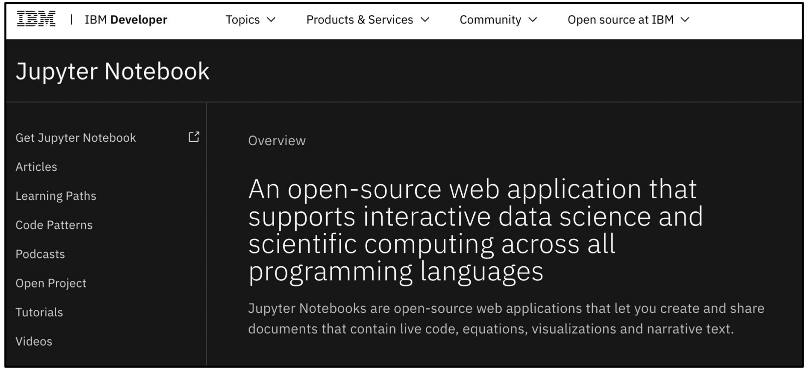
A multi-user version of the notebook designed for companies, classrooms and research labs

binder

Analysis platforms/environments like Jupyter grow in popularity and started to be some of the “standard” in several domains of data analysis

Here some examples of such tools, where Jupyter is offer as a service by public or private institutions

Analysis code SaaS

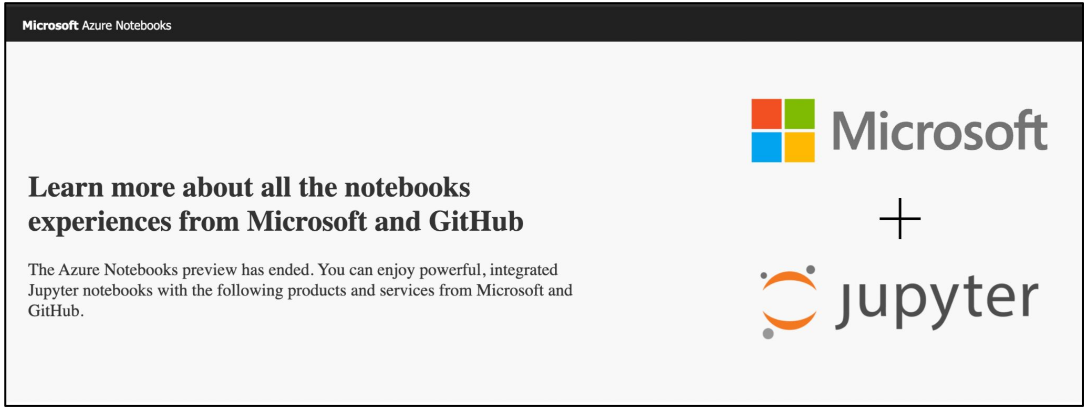


Jupyter Notebook

Overview

An open-source web application that supports interactive data science and scientific computing across all programming languages

Jupyter Notebooks are open-source web applications that let you create and share documents that contain live code, equations, visualizations and narrative text.



Learn more about all the notebooks experiences from Microsoft and GitHub

The Azure Notebooks preview has ended. You can enjoy powerful, integrated Jupyter notebooks with the following products and services from Microsoft and GitHub.

A multi-user version of the notebook designed for companies, classrooms and research labs

ATLASSIAN
Bitbucket

GitLab

GitHub

Git +
CI/CD

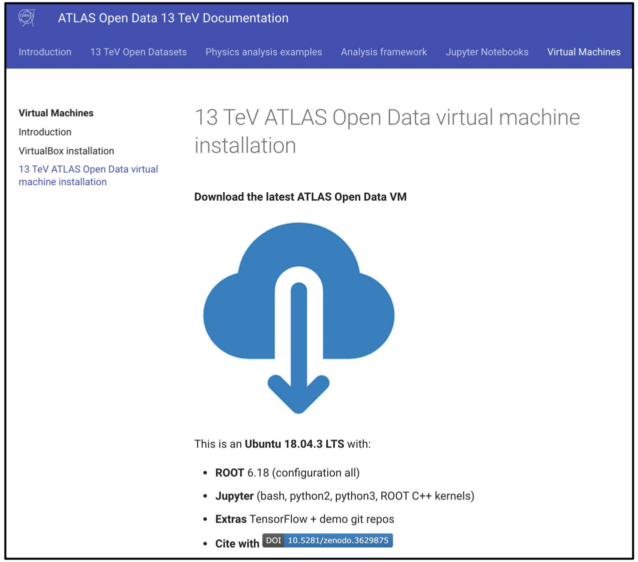
There are several companies that allow the creation and hosting of Git repositories (you are using one of those right now)

But you can also self-hosted one of those instances.
They also come with a lot of functionalities like CI/CD

The computing infrastructure is, for example, your laptop/desktop machine. There the needed OS, software and tools are installed to perform the analysis (like what we are doing during the school)

But you can also get the needed environment using Virtual Machines or containers

Computing IaaS



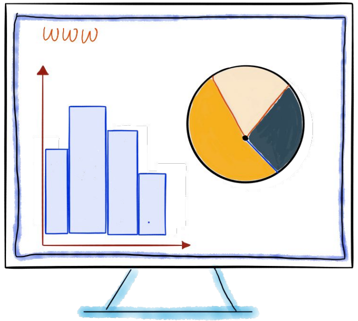
Virtual Machines
Introduction
VirtualBox installation
13 TeV ATLAS Open Data virtual machine installation

13 TeV ATLAS Open Data virtual machine installation

Download the latest ATLAS Open Data VM

This is an **Ubuntu 18.04.3 LTS** with:

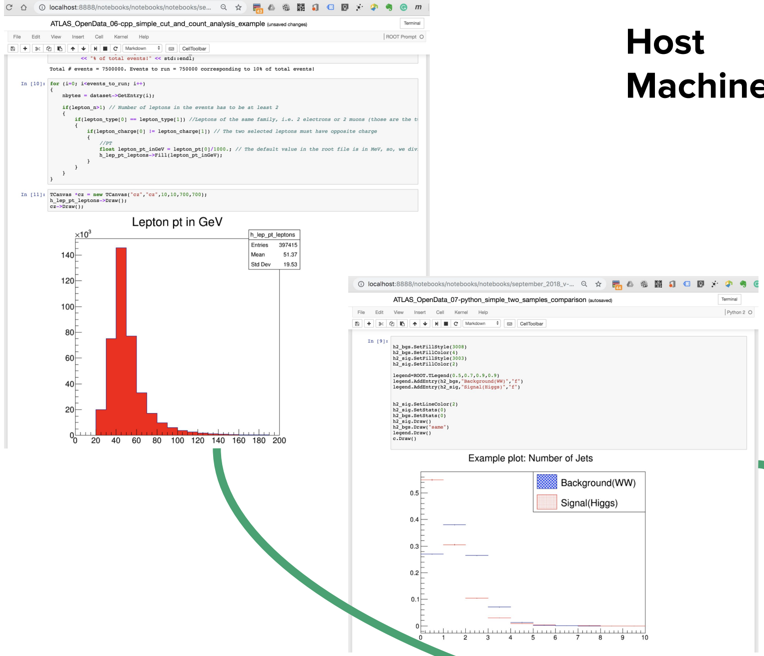
- **ROOT** 6.18 (configuration all)
- **Jupyter** (bash, python2, python3, ROOT C++ kernels)
- **Extras** TensorFlow + demo git repos
- **Cite with** DOI 10.5281/zenodo.3629875



binder

Examples of this VM usage

Host Machine



localhost:8888/notebooks/notebooks/notebooks/se...

ATLAS_OpenData_06-cpp_simple_cut_and_count_analysis_example (unsaved changes)

```
          << "% of total events!" << std::endl;
```

```
Total # events = 7500000. Events to run = 750000 corresponding to 10% of total events!
```

In [10]:

```
for (i=0; i<events_to_run; i++)
{
    nbytes = dataset->GetEntry(i);

    if(lepton_n>1) // Number of leptons in the events has to be at least 2
    {
        if(lepton_type[0] == lepton_type[1]) //Leptons of the same family, i.e. 2 electrons or 2 muons (those are the t
        {
            if(lepton_charge[0] != lepton_charge[1]) // The two selected leptons must have opposite charge
            {
                //PT
                float lepton_pt_inGeV = lepton_pt[0]/1000.; // The default value in the root file is in MeV, so, we div
                h_lep_pt_leptons->Fill(lepton_pt_inGeV);
            }
        }
    }
}
```

In [11]:

```
TCanvas *cz = new TCanvas("cz","cz",10,10,700,700);
h_lep_pt_leptons->Draw();
cz->Draw();
```

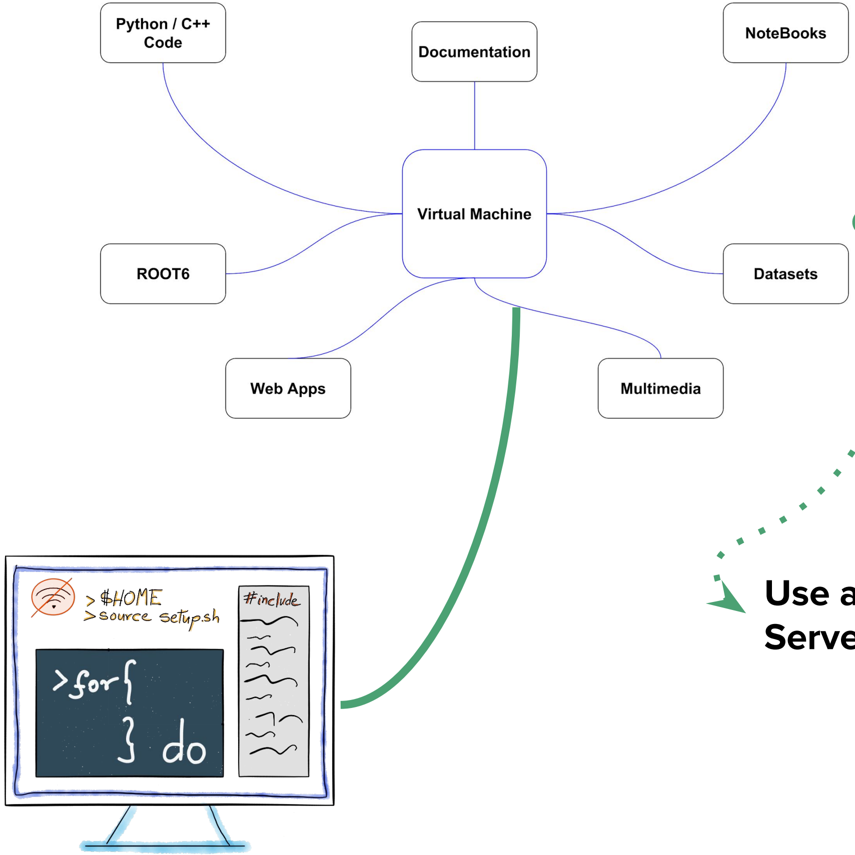
h_lep_pt_leptons	
Entries	397415
Mean	51.37
Std Dev	19.53

localhost:8888/notebooks/notebooks/notebooks/september_2018_v-...

ATLAS_OpenData_07-python_simple_two_samples_comparison (autosaved)

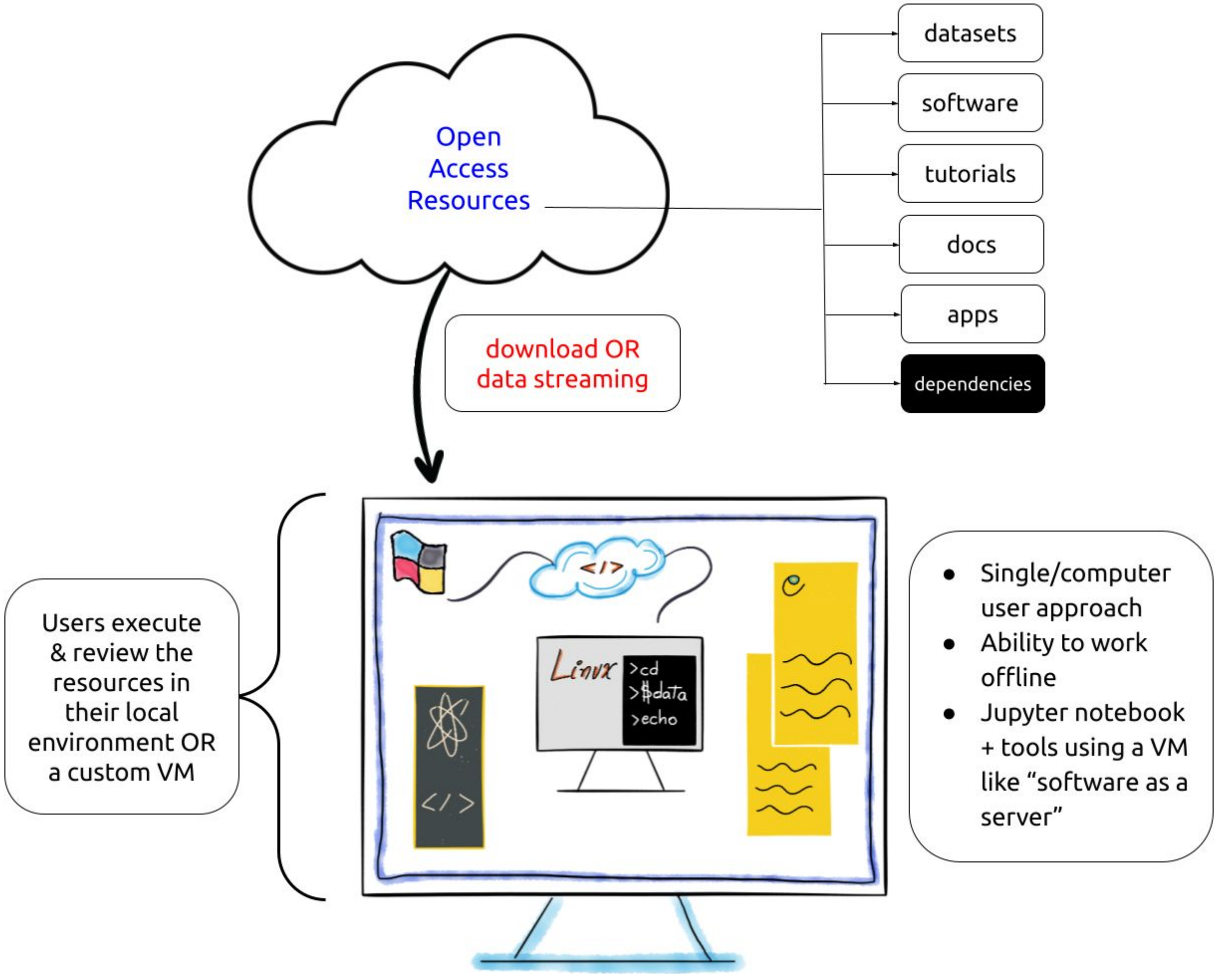
In [9]:

```
h2_bgs.SetFillStyle(3008)
h2_bgs.SetFillColor(4)
h2_sig.SetFillStyle(3003)
h2_sig.SetFillColor(2)

legend=ROOT.TLegend(0.5,0.7,0.9,0.9)
legend.AddEntry(h2_bgs,"Background(WW)","f")
legend.AddEntry(h2_sig,"Signal(Higgs)","f")


h2_sig.SetLineColor(2)
h2_sig.SetStats(0)
h2_bgs.SetStats(0)
h2_sig.Draw()
h2_bgs.Draw("same")
legend.Draw()
c.Draw()
```



Use as a Server

A classical example is to host a VM that we can use as a private “server” isolating and preserving the working environment



Open Access Resources

- datasets
- software
- tutorials
- docs
- apps
- dependencies

download OR data streaming

Users execute & review the resources in their local environment OR a custom VM

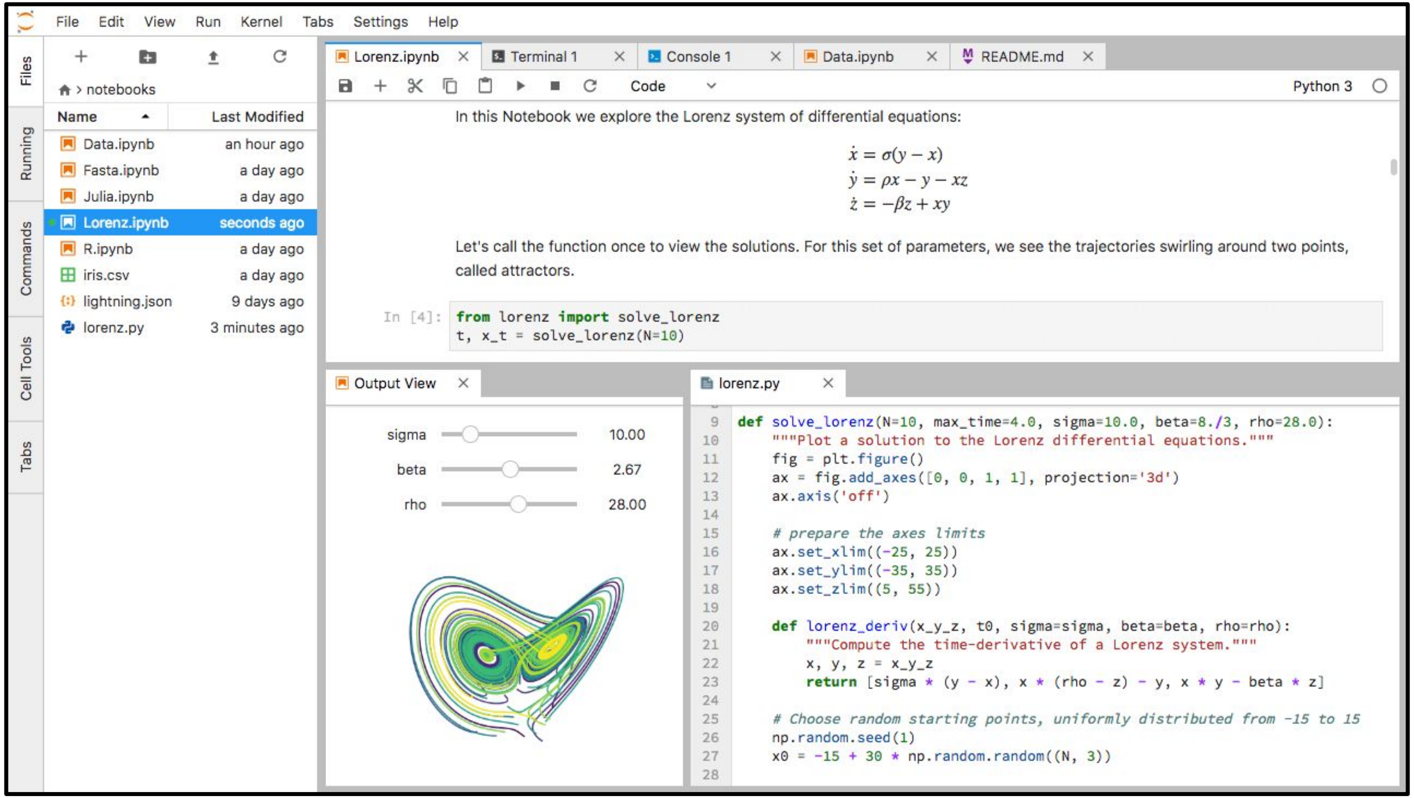
- Single/computer user approach
- Ability to work offline
- Jupyter notebook + tools using a VM like “software as a server”

The JupyterLab **UI**

The JupyterLab UI

A well-known tool for all of us (data analysis and visualisation) is the Jupyter notebook.

JupyterLab is a suite of tools and features that allow interacting with multiple elements in a single view. And do the computation, of course. https://jupyterlab.readthedocs.io/en/stable/



Name	Last Modified
Data.ipynb	an hour ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ipynb	seconds ago
R.ipynb	a day ago
iris.csv	a day ago
lightning.json	9 days ago
lorenz.py	3 minutes ago

In this Notebook we explore the Lorenz system of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

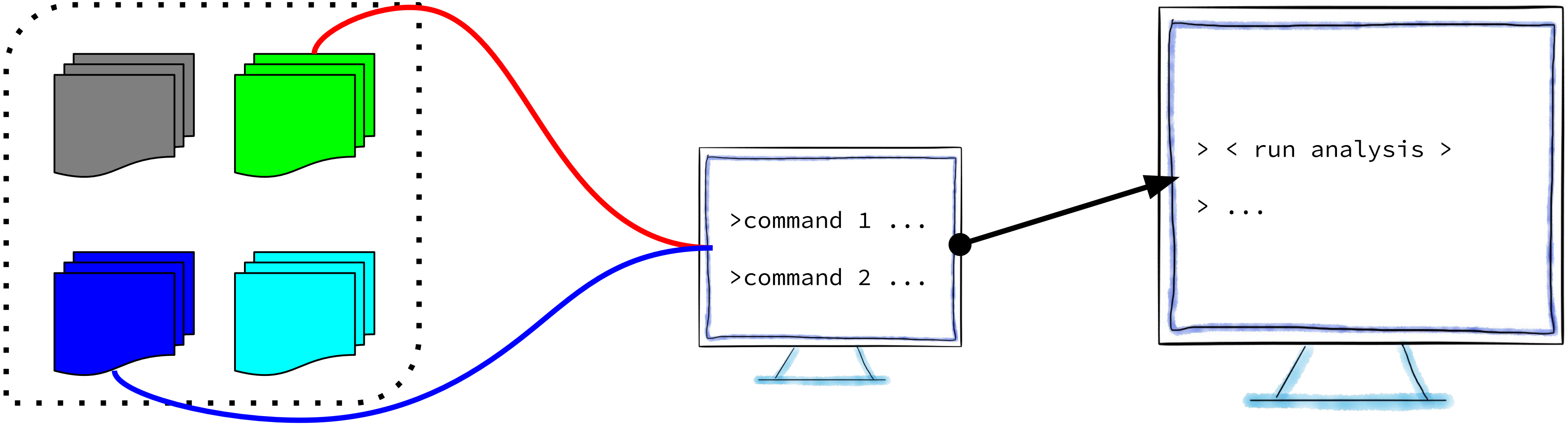
```
In [4]: from lorenz import solve_lorenz
        t, x_t = solve_lorenz(N=10)
```

sigma 10.00

beta 2.67

rho 28.00

```
def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
    """Plot a solution to the Lorenz differential equations."""
    fig = plt.figure()
    ax = fig.add_axes([0, 0, 1, 1], projection='3d')
    ax.axis('off')

    # prepare the axes limits
    ax.set_xlim((-25, 25))
    ax.set_ylim((-35, 35))
    ax.set_zlim((5, 55))

    def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
        """Compute the time-derivative of a Lorenz system."""
        x, y, z = x_y_z
        return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]

    # Choose random starting points, uniformly distributed from -15 to 15
    np.random.seed(1)
    x0 = -15 + 30 * np.random.random((N, 3))

```



```
>command 1 ...
>command 2 ...
```

```
> < run analysis >
> ...
```

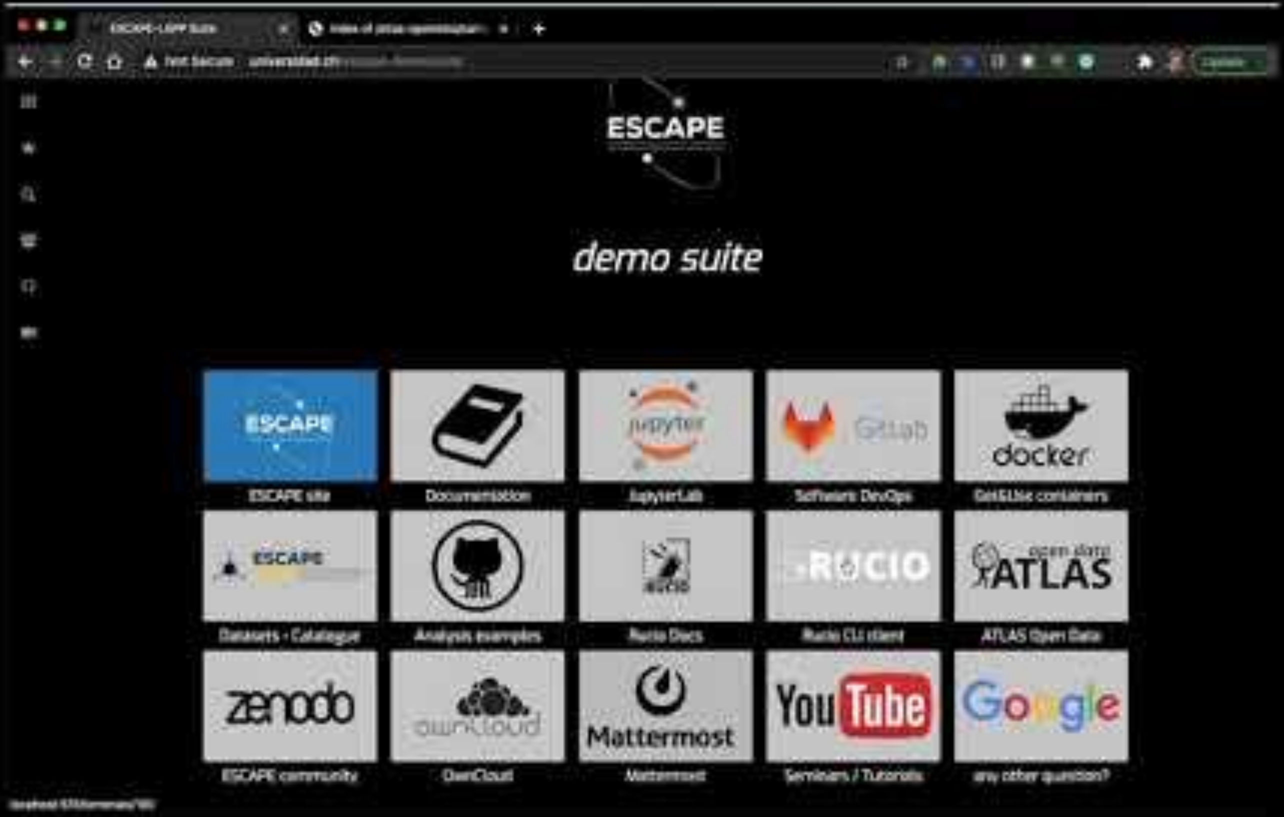
Repositories and some stored datasets

Computer facility

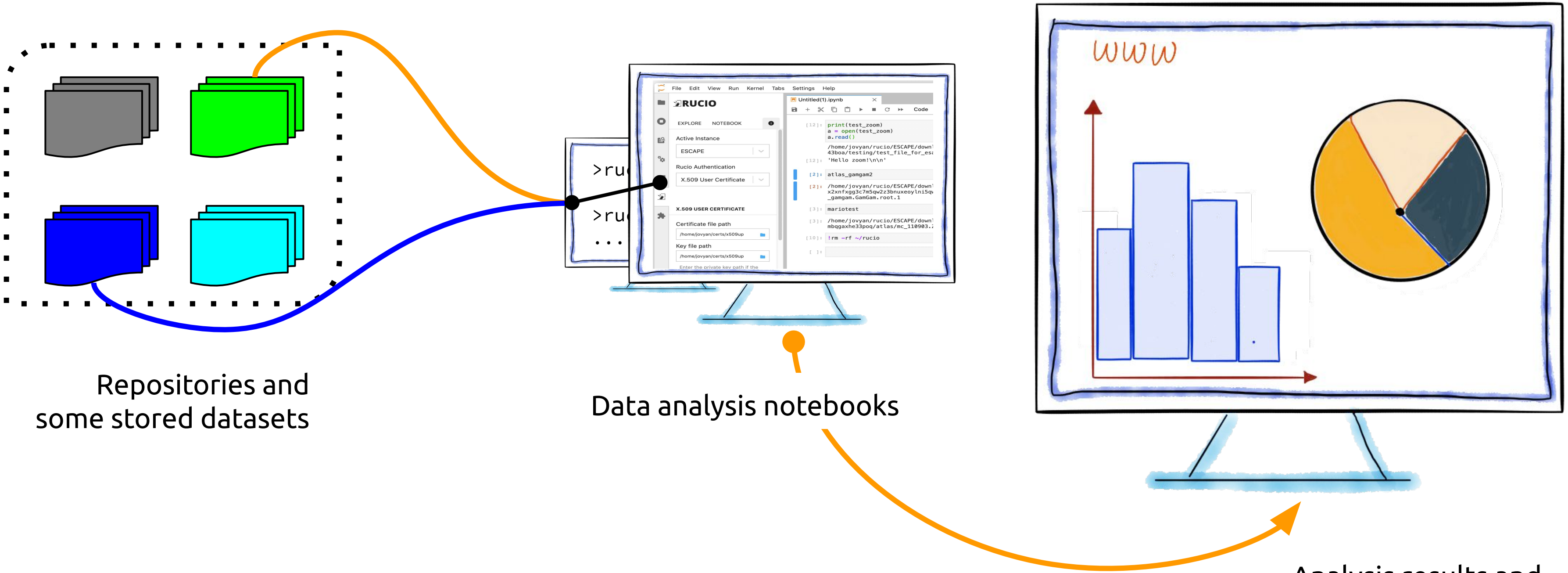
Analysis development

An example of JupyterLab

(a 90 sec video)



http://universidad.ch/escape-demo/suite/



Active Instance

ESCAPE

Rucio Authentication

X.509 User Certificate

X.509 USER CERTIFICATE

Certificate file path

/home/jovyan/certs/x509up

Key file path

/home/jovyan/certs/x509up

Enter the private key path if the

```
[12]: print(test_zoom)
      a = open(test_zoom)
      a.read()
      /home/jovyan/rucio/ESCAPE/downl
      43boa/testing/test_file_for_esa
[12]: 'Hello zoom!\n\n'

[2]:  atlas_gamgam2
[2]:  /home/jovyan/rucio/ESCAPE/downl
      x2xnfxgg3c7m5qw2z3bnuxeoylni5qw
      _gamgam.GamGam.root.1
[3]:  mariotest
[3]:  /home/jovyan/rucio/ESCAPE/downl
      mbqgaxhe33poq/atlas/mc_110903.2
[10]: !rm -rf ~/rucio
[ ]:
```

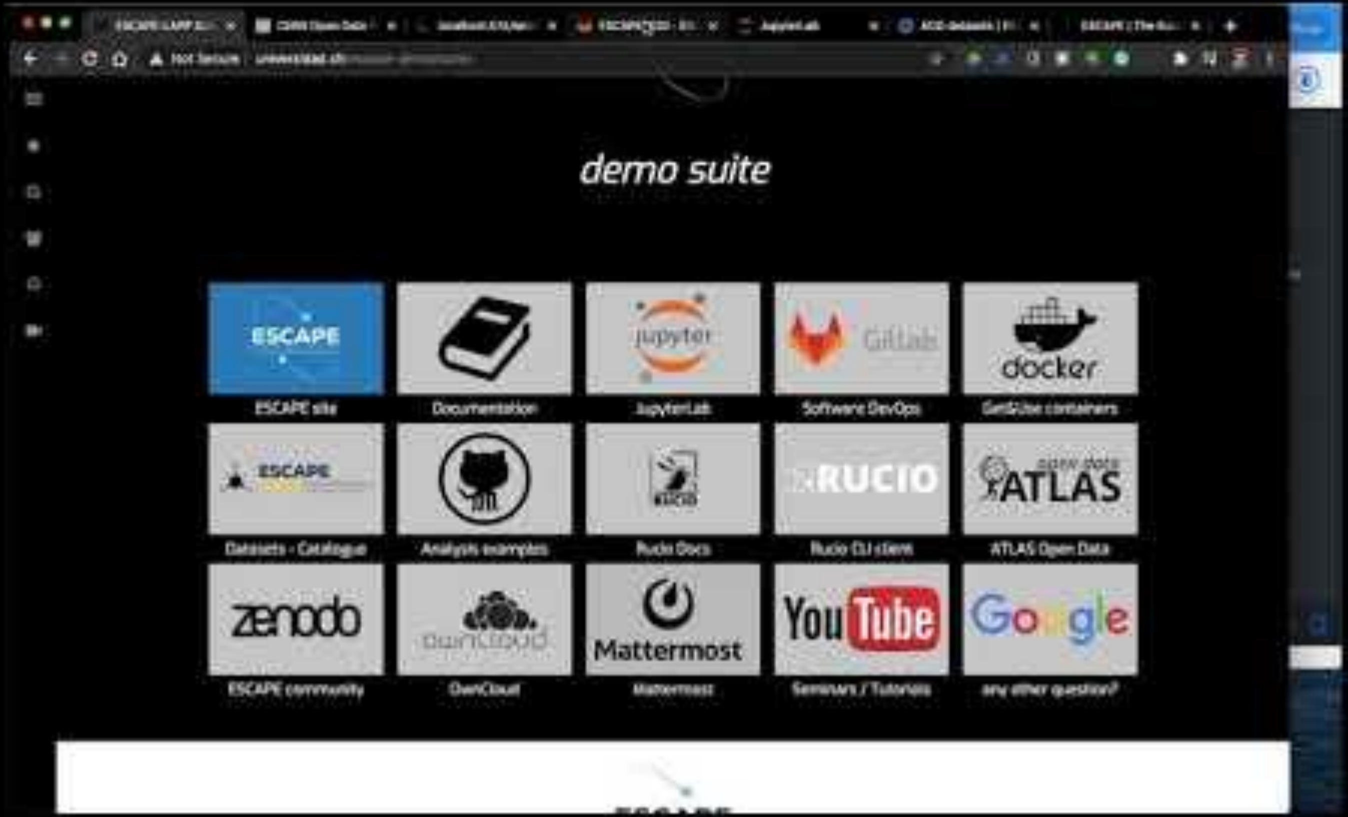
Repositories and
some stored datasets

Data analysis notebooks

Analysis results and
visualisation

An example of JupyterLab

(a 150 sec video)



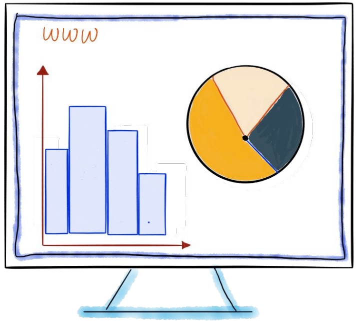
http://universidad.ch/escape-demo/suite/

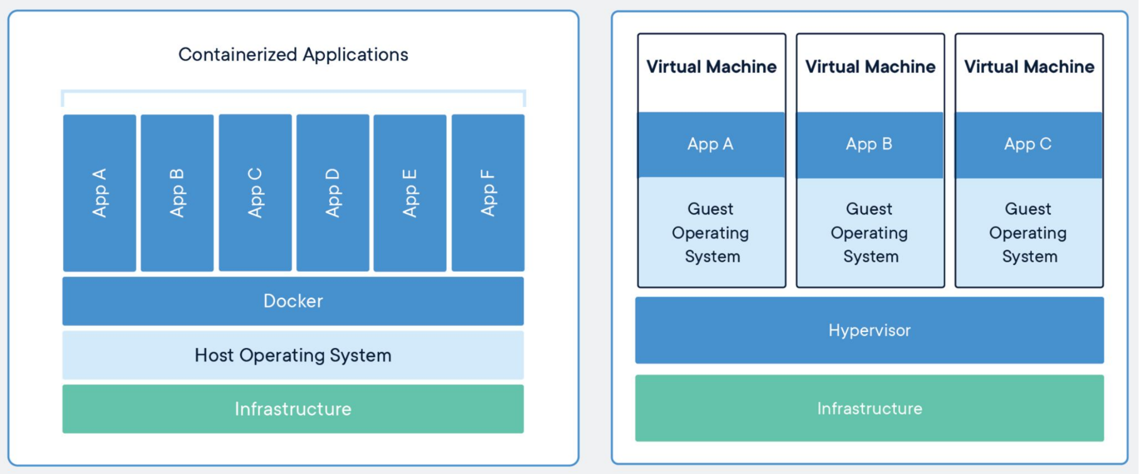
Containers using **Docker**

Containers allow the preservation and reproducibility of software environments and applications

As an example, we can have a Docker container and execute it in our machine or in a remote cloud… inside a VM :)

Computing IaaS

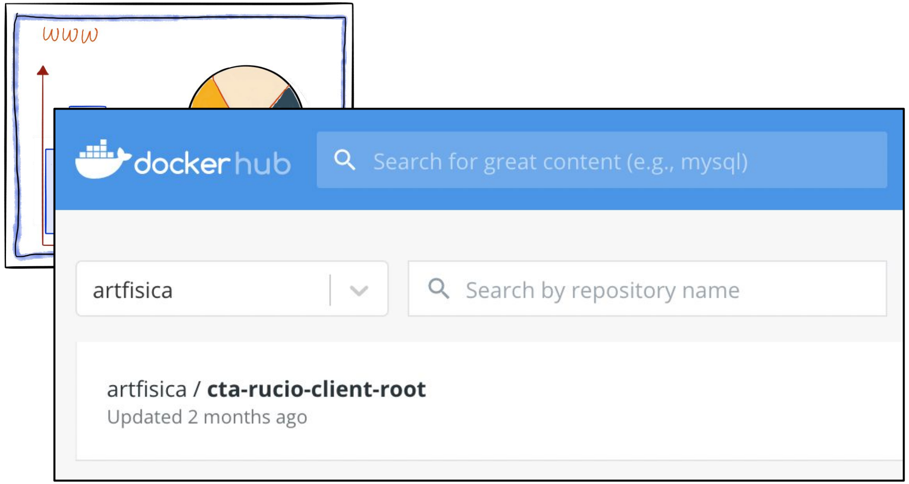




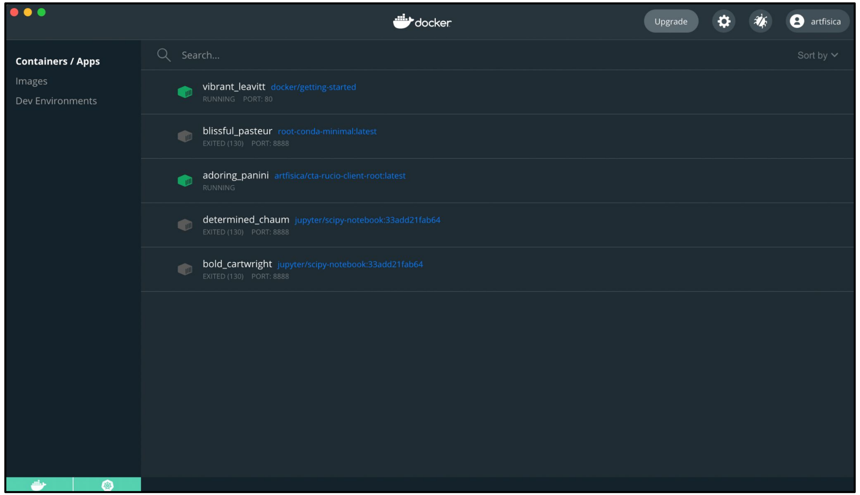
Containers allow the preservation and reproducibility of software environments and applications

As an example, we can have a Docker container and execute it in our machine or in a remote cloud… inside a VM :)

Computing IaaS


www
docker hub
Search for great content (e.g., mysql)
artfisica
Search by repository name

artfisica / **cta-rucio-client-root**
Updated 2 months ago



vibrant_leavitt docker/getting-started
RUNNING PORT: 80

blissful_pasteur root-conda-minimal:latest
EXITED (130) PORT: 8888

adoring_panini artfisica/cta-rucio-client-root:latest
RUNNING

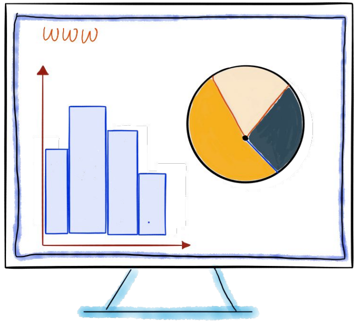
determined_chaum jupyter/scipy-notebook:33add21fab64
EXITED (130) PORT: 8888

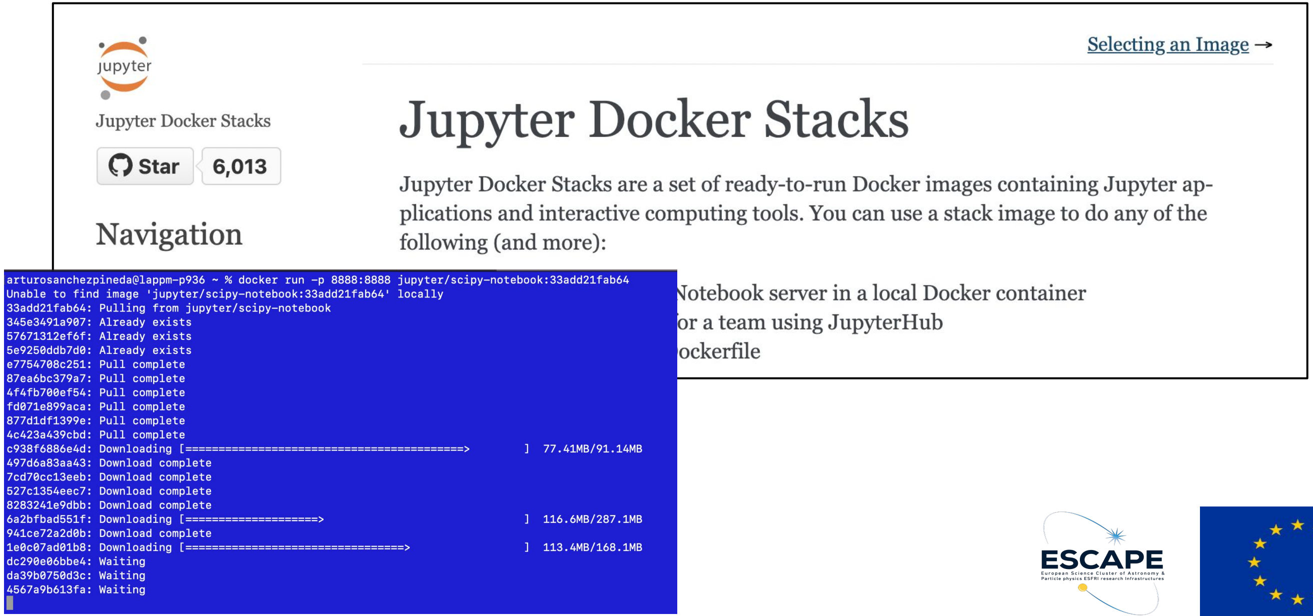
bold_cartwright jupyter/scipy-notebook:33add21fab64
EXITED (130) PORT: 8888

For example, you can use a Docker container to deploy a JupyterLab instance in your computer

Same can be done with JupyterHub

Computing
IaaS





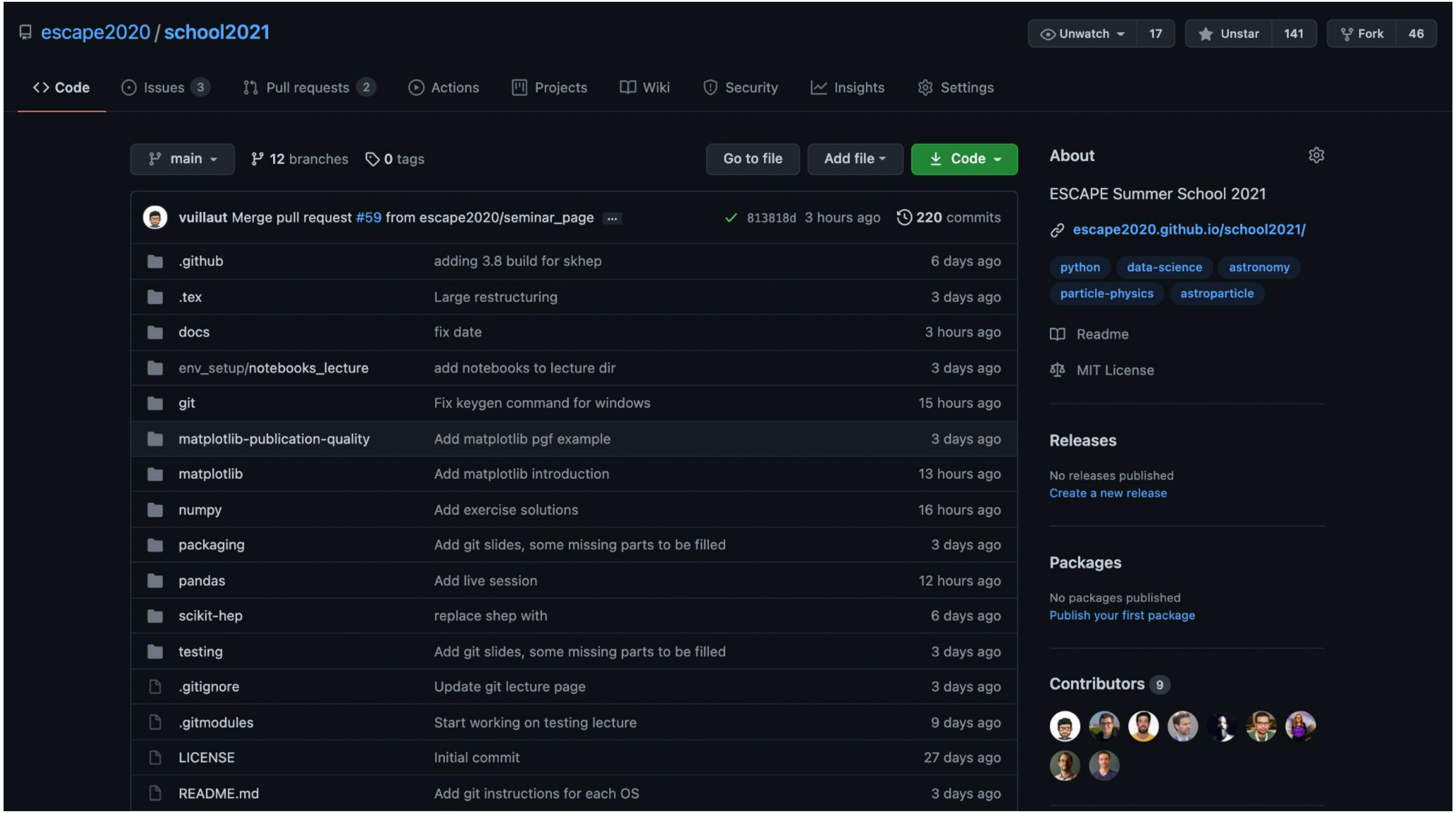
Jupyter Docker Stacks

Jupyter Docker Stacks are a set of ready-to-run Docker images containing Jupyter applications and interactive computing tools. You can use a stack image to do any of the following (and more):

Notebook server in a local Docker container
or a team using JupyterHub
ockerfile

```
arturosanchezpineda@lappm-p936 ~ % docker run -p 8888:8888 jupyter/scipy-notebook:33add21fab64
Unable to find image 'jupyter/scipy-notebook:33add21fab64' locally
33add21fab64: Pulling from jupyter/scipy-notebook
345e3491a907: Already exists
57671312ef6f: Already exists
5e9250ddb7d0: Already exists
e7754708c251: Pull complete
87ea6bc379a7: Pull complete
4f4fb700ef54: Pull complete
fd071e899aca: Pull complete
877d1df1399e: Pull complete
4c423a439cbd: Pull complete
c938f6886e4d: Downloading [==========================================>       ]  77.41MB/91.14MB
497d6a83aa43: Download complete
7cd70cc13eeb: Download complete
527c1354eec7: Download complete
8283241e9dbb: Download complete
6a2bfbad551f: Downloading [====================>                             ] 116.6MB/287.1MB
941ce72a2d0b: Download complete
1e0c07ad01b8: Downloading [=================================>                ] 113.4MB/168.1MB
dc290e06bbe4: Waiting
da39b0750d3c: Waiting
4567a9b613fa: Waiting
```

Installing Docker and **reproduce notebooks**



escape2020 / school2021

vuillaut Merge pull request #59 from escape2020/seminar_page ... ✓ 813818d 3 hours ago 220 commits

.github	adding 3.8 build for skhep	6 days ago
.tex	Large restructuring	3 days ago
docs	fix date	3 hours ago
env_setup/notebooks_lecture	add notebooks to lecture dir	3 days ago
git	Fix keygen command for windows	15 hours ago
matplotlib-publication-quality	Add matplotlib pgf example	3 days ago
matplotlib	Add matplotlib introduction	13 hours ago
numpy	Add exercise solutions	16 hours ago
packaging	Add git slides, some missing parts to be filled	3 days ago
pandas	Add live session	12 hours ago
scikit-hep	replace shep with	6 days ago
testing	Add git slides, some missing parts to be filled	3 days ago
.gitignore	Update git lecture page	3 days ago
.gitmodules	Start working on testing lecture	9 days ago
LICENSE	Initial commit	27 days ago
README.md	Add git instructions for each OS	3 days ago

About

ESCAPE Summer School 2021

escape2020.github.io/school2021/

python data-science astronomy particle-physics astroparticle

Readme

MIT License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 9

A concrete example **reproducible analysis**

reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.

Scalable

Support for remote compute clouds.

Reusable

Containerise once, reuse elsewhere. Cloud-native.

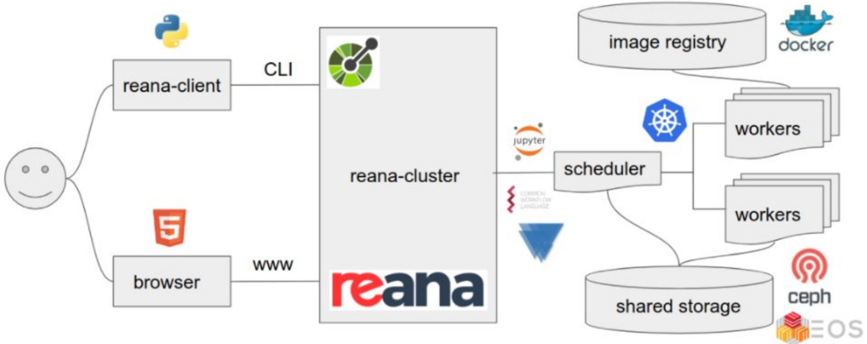
Free

Free Software. MIT licence. Made with ❤ at CERN.

https://reanahub.io/

Reproducible analyses

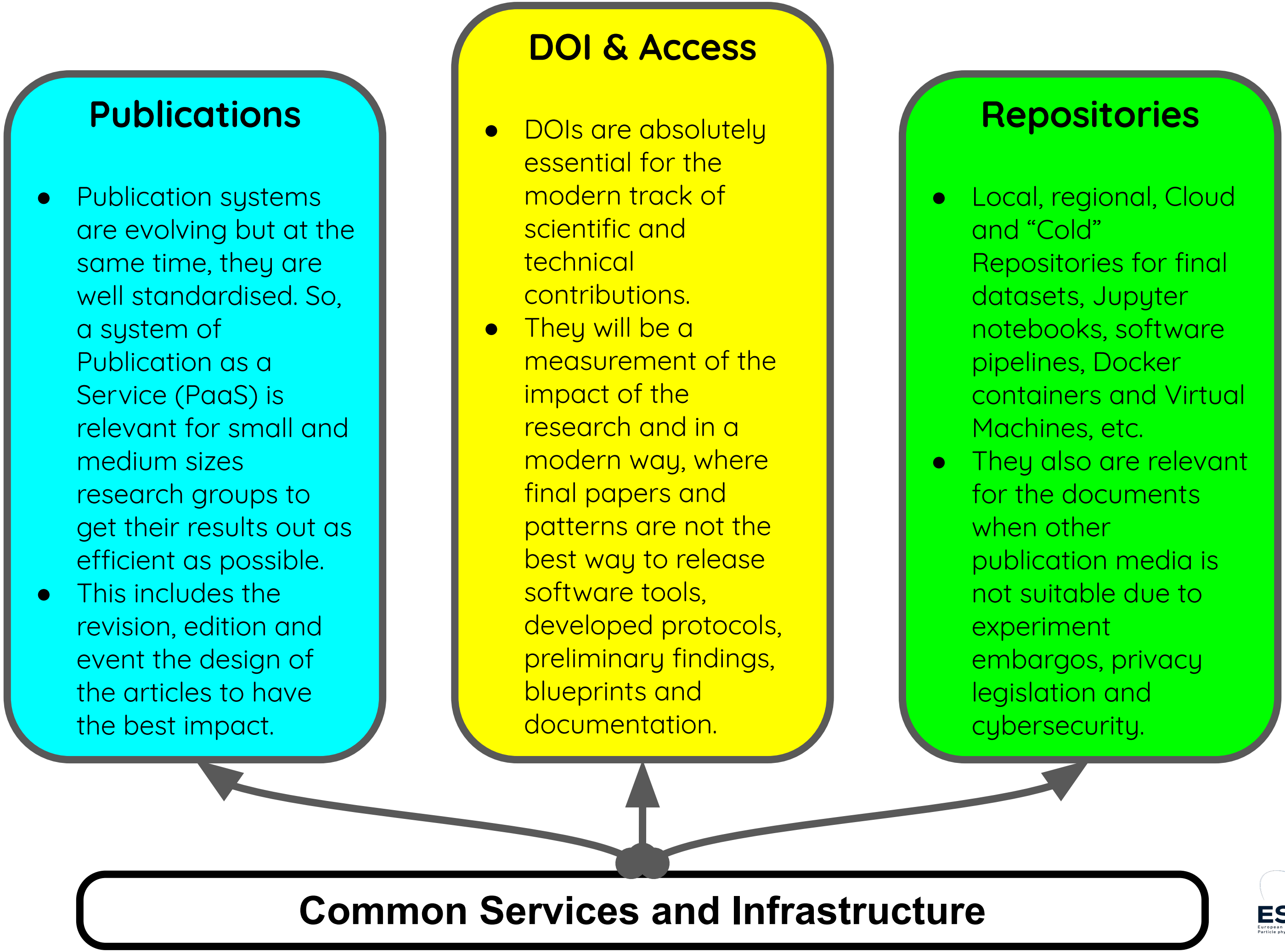
Lesson on reproducible analyses and reusable containerised scientific workflows



reana

https://awesome-workshop.github.io/reproducible-analyses/

Last **comments**



Publications

- Publication systems are evolving but at the same time, they are well standardised. So, a system of Publication as a Service (PaaS) is relevant for small and medium sizes research groups to get their results out as efficient as possible.
- This includes the revision, edition and event the design of the articles to have the best impact.

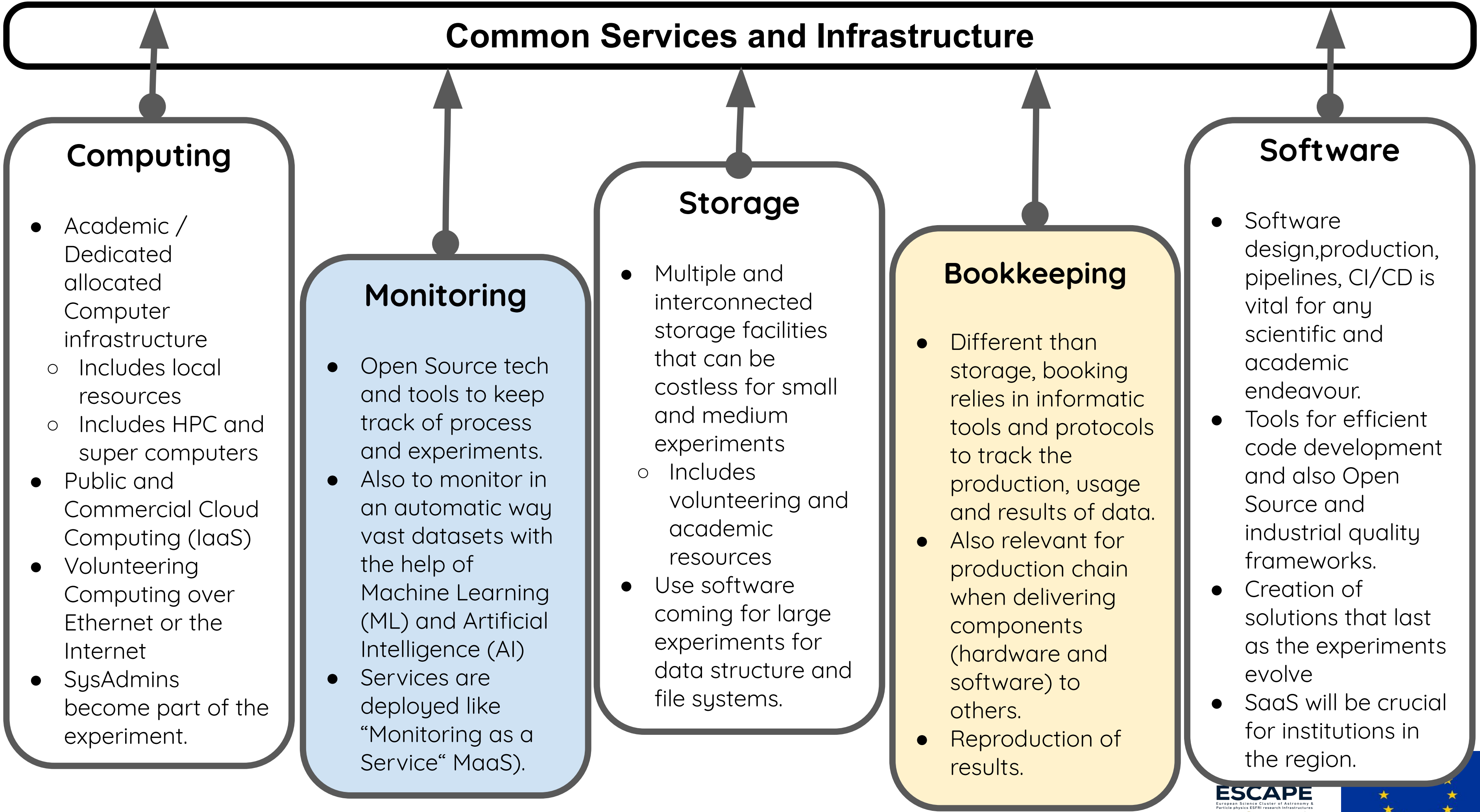
DOI & Access

- DOIs are absolutely essential for the modern track of scientific and technical contributions.
- They will be a measurement of the impact of the research and in a modern way, where final papers and patterns are not the best way to release software tools, developed protocols, preliminary findings, blueprints and documentation.

Repositories

- Local, regional, Cloud and “Cold” Repositories for final datasets, Jupyter notebooks, software pipelines, Docker containers and Virtual Machines, etc.
- They also are relevant for the documents when other publication media is not suitable due to experiment embargos, privacy legislation and cybersecurity.

Common Services and Infrastructure



Common Services and Infrastructure

Computing

- Academic / Dedicated allocated Computer infrastructure
  - Includes local resources
  - Includes HPC and super computers
- Public and Commercial Cloud Computing (IaaS)
- Volunteering Computing over Ethernet or the Internet
- SysAdmins become part of the experiment.

Monitoring

- Open Source tech and tools to keep track of process and experiments.
- Also to monitor in an automatic way vast datasets with the help of Machine Learning (ML) and Artificial Intelligence (AI)
- Services are deployed like “Monitoring as a Service“ MaaS).

Storage

- Multiple and interconnected storage facilities that can be costless for small and medium experiments
  - Includes volunteering and academic resources
- Use software coming for large experiments for data structure and file systems.

Bookkeeping

- Different than storage, booking relies in informatic tools and protocols to track the production, usage and results of data.
- Also relevant for production chain when delivering components (hardware and software) to others.
- Reproduction of results.

Software

- Software design,production, pipelines, CI/CD is vital for any scientific and academic endeavour.
- Tools for efficient code development and also Open Source and industrial quality frameworks.
- Creation of solutions that last as the experiments evolve
- SaaS will be crucial for institutions in the region.

In my view, **reproducibility** refers to a series of principles, techniques, tools and practical considerations that allow the documentation, recording and preservation of data analysis pipelines — enhancing the possibilities of collaborations across borders and increasing the probabilities of replicating results by others (and yourself) in the future.

Reproducibility involves using standard and well-established protocols to ensure that your code will survive outside your computer, the passing of time and that others will be able to use it as a starting point for new analysis.

Thanks!

Arturo Sánchez Pineda

arturos@cern.ch

Linkedin
/arturo-sanchez-pineda/

@artfisica

https://twitter.com/Arturo_RSP

I am post-doctoral fellow at LAPP-CNRS, France. Member of the ESCAPE and ATLAS groups.
I studied Fundamental Physics and System Engineering in the Universidad de Los Andes, Venezuela, with a PhD in Fundamental and Applied Physics from Università di Napoli "Federico II", Italy.
I was previously a postdoctoral fellow at Physics Department at Università di Udine and an Associate at INFN, Italy. Also, an ATLAS TDAQ System Administrator at CERN, Switzerland, and Research Associate at the High Energy, Cosmology and Astroparticle Section at ICTP, Italy.
← **And I do a lot of outreach :)**