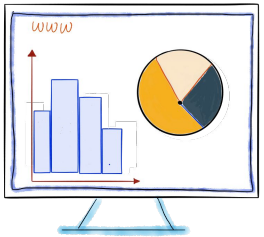
A proposal for Open Access data and tools multi-user deployment using ATLAS Open Data for Education



Arturo Sánchez Pineda (LAPP) & Giovanni Guerrieri (Udine)
on behalf of the ATLAS Open Data and Tools team

May 2021 - vCHEP conference

arturos@cern.ch

Overview: Making the case

- Software as a Service (SaaS) and Infrastructure as a Service (IaaS)* have reshaped the way of data handling, analysis, storage, and sharing; particularly in multinational collaborations
- We explore how a SaaS + IaaS approach can be adapted to **modest scenarios and institutions**, using virtual machines and containers, **for educational purposes**
- The target audience of this products proposal are **trainers and small/medium institutions SysAdmin**
- To explore this idea we are using the current ATLAS Open Data (OD) and analysis examples
- **A couple of prototypes are in place.** They are based on terraform.io for cloud instance creation, and docker containers to obtain Jupyter{Lab,Hub}-based environments for multi-user scenarios

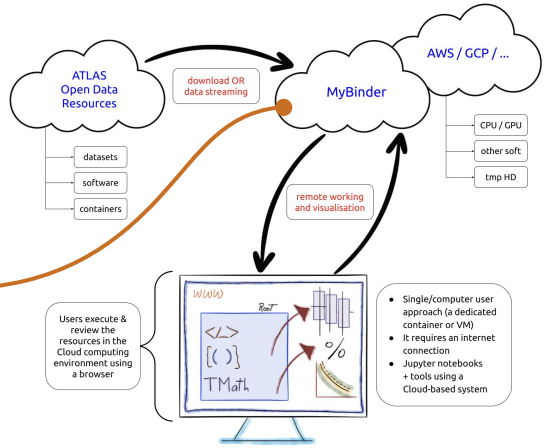
*or Infrastructure as Code (IaC)

The ATLAS OD users and settings

Institutional Infrastructure & cloud

Educators use ATLAS OD resources to complement diverse HEP and data-analysis training programs

They use: * institutional resources (e.g. in-house computers), * free (e.g. MyBinder, CoLab) or * commercial cloud (e.g. AWS, GCP...) to run hands-on sessions

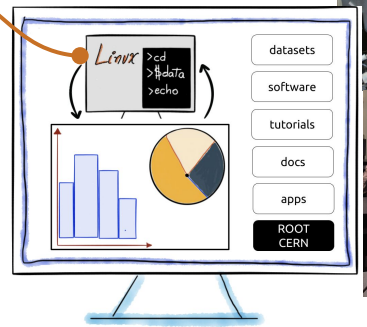


ATLAS OD single-machine tools

Students bring their computers and use a VM to run long-term projects, like a thesis or a university course, based on or profiting from a Jupyter UI

Similar cases when running online sessions

In all those cases, a simple setup is vital



Black lines for illustration purposes only. Visit https://atlas.cern/discover/collaboration

A concrete proposal: IaaS and SaaS

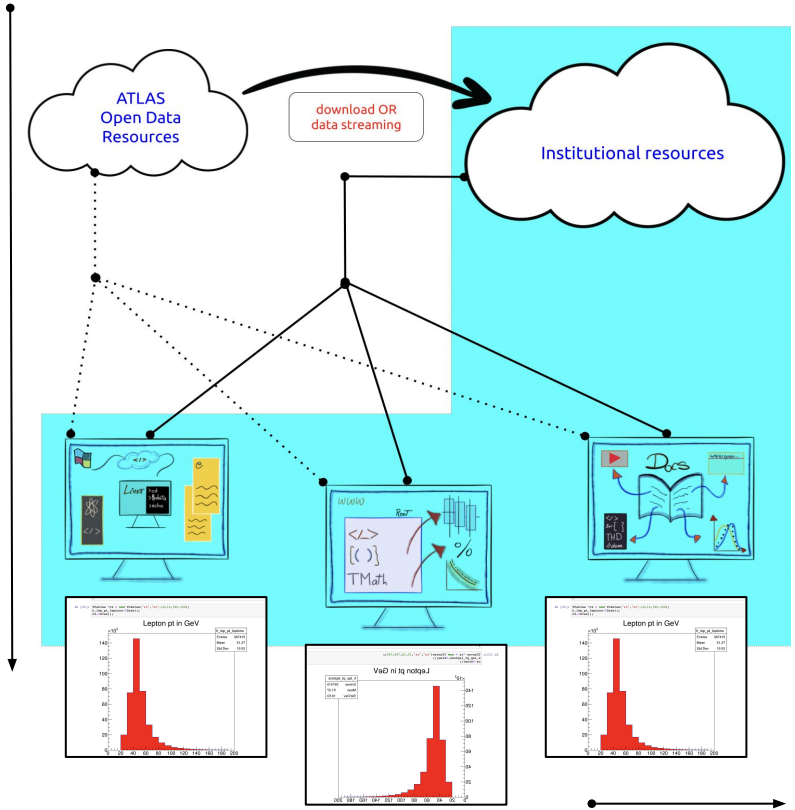
A

We are designing, creating, testing (and will release) a series of:

- **Terraform IaC recipes ready to use in cloud infrastructure**
  Current development uses the CERN OpenStack. Also, working on prototypes for AWS and GCP instances

- **Containers that can replace or enhance the existing ATLAS OD single-user VM**
  Preliminary container solutions tested using DockerHub

Combinations of containers using JupyterHub Docker Spawner to deploy multiple-user JupyterHub with same single-user configuration



B

Users can deploy ATLAS OD resources in a cloud:

* On institutional resources out of the HEP community and with limited SysAdmin,
* or commercial clouds rented by institutions or individuals running short-term workshops

We explore the idea of:
Using the same single-user container in such a multi-user environment, making possible to have a "single" product that fits multiple scenarios

Current prototype: SaaS using Docker

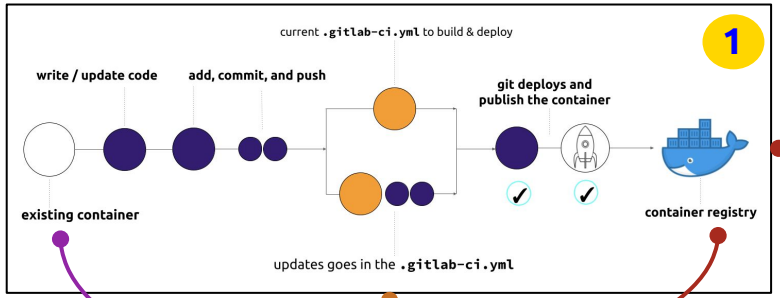
The containers (SaaS) have the tools needed to perform ATLAS OD and other HEP educational analyses

1 → Start with the development, test and deployment of the containers, making use of the GitLab CI/CD infrastructure at CERN

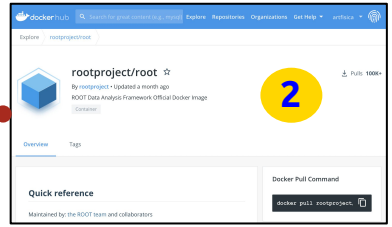
2→ The containers are deployed in a register

3→ The ATLAS OD website is used to document those in a friendly UI

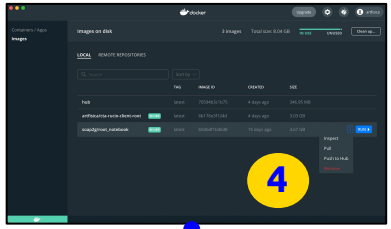
4→ At the user level: containers are managed with Docker app, hosting and using several environment (analogous to use VMs+VirtualBox)

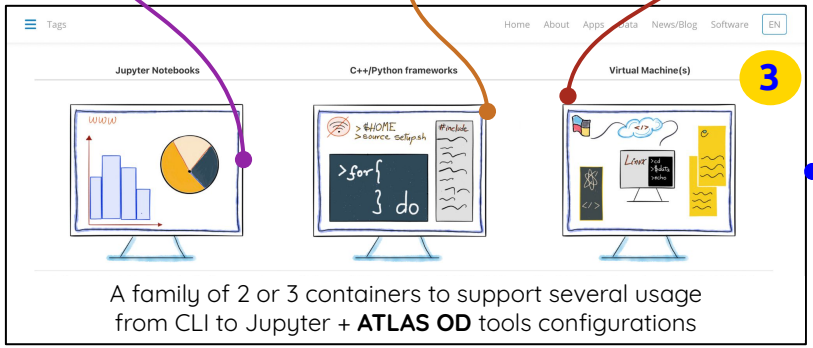


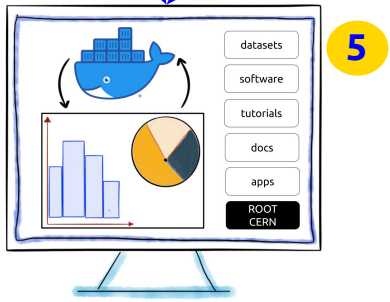
(example, CERN-ROOT for illustration only)



5→ Users run and modify the ATLAS OD examples

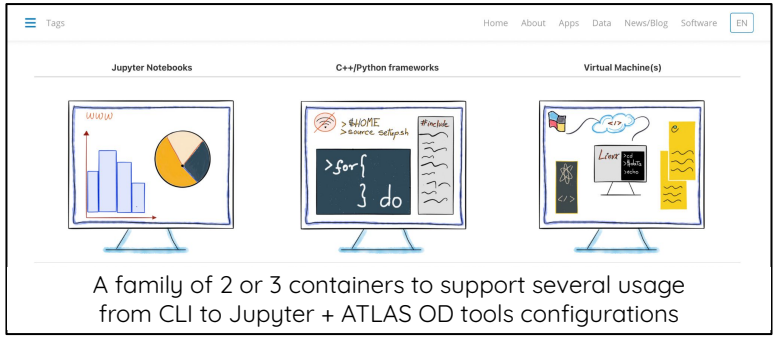






Leaving behind the less flexible VirtualBox + VM model

A dedicated hub for containers and recipes

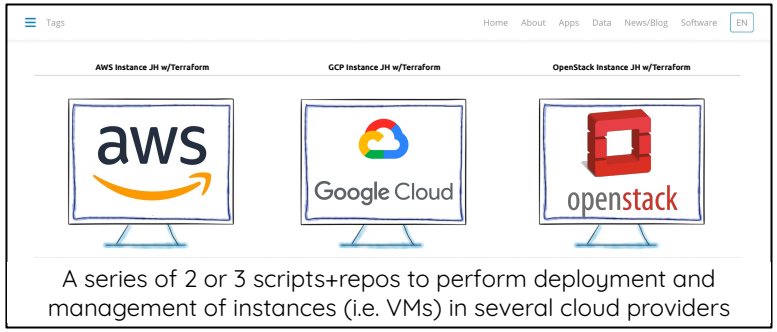


A family of 2 or 3 containers to support several usage from CLI to Jupyter + ATLAS OD tools configurations

Solutions are designed to work standalone

+

Or in combination for a complete IaC + data analysis suite



A series of 2 or 3 scripts+repos to perform deployment and management of instances (i.e. VMs) in several cloud providers

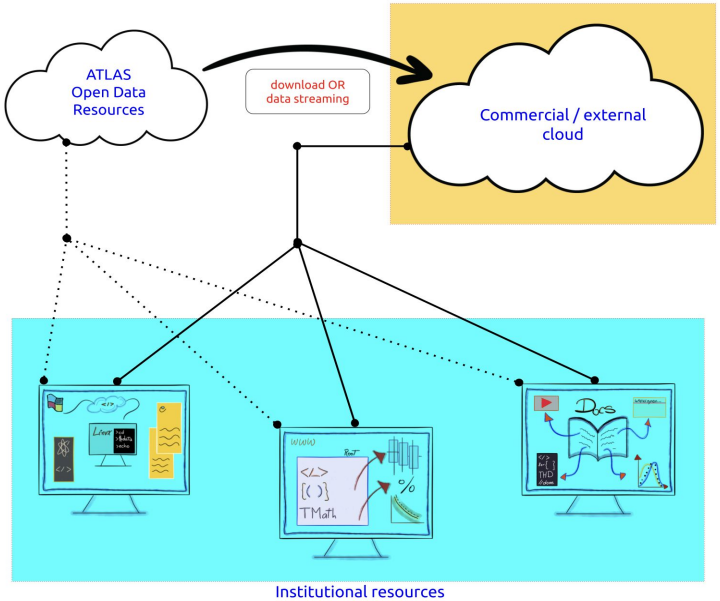
The combination IaaS + SaaS comes when, after the instance creation, it uses containers for the environment configuration, **approaching *-asymptotically-* to a "single-click" solution** deployment **for small and medium-size institutions or individual trainers**

- A proof-of-concept was performed using Terraform and the CERN OpenStack cloud last winter. Also, using a commercial cloud free-trial account
- New containers have been created, and the JupyterHub Docker Spawner technique is helping to develop a flexible pipeline. Allowing to design a small collection of containers that we can seamlessly use in a single-user mode (e.g. a laptop) or a multi-user solution like a JupyterHub

Summary and future

This proposal shows a path on how we can deploy reproducible educational data-analysis platforms at small and medium project/institutions, taking advantage of current and widely available software and computing tools

- We intend to deliver a series of recipes and repositories that can be used by educators and IT service providers to deploy IaaS and SaaS on-premises or in commercial clouds
- Finally, we plan to deliver the first production-ready recipes and containers on the ATLAS Open Data website by the end of 2021.



Backup

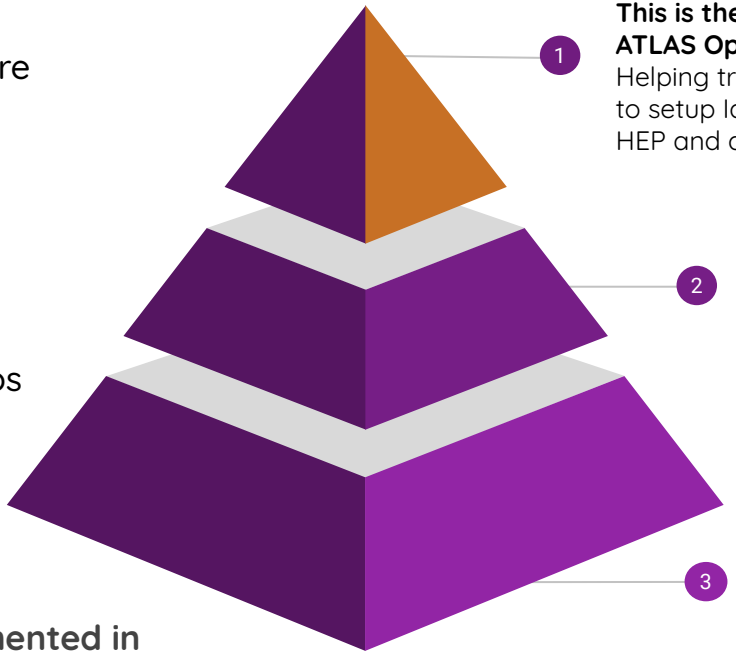
arturos@cern.ch

Target audience of the project

We aim to develop and deploy a series of tested and production-ready recipes to deploy IaaS and SaaS in small infrastructure

- meaning IaaS based on terraform.io for single-VM creation on clouds
- docker containers to obtain SaaS Jupyter{Lab,Hub}-based environments for multi-user scenarios

Using **ATLAS OD** resources for educational activities



Trainers and small/medium SysAdmins institutions

This is the target audience of this ATLAS Open Data products proposal
Helping trainers and small institutions to setup IaaS and SaaS for teaching HEP and data sciences

University students: Master and small PhDs
Main target audience of current ATLAS Open Data resources

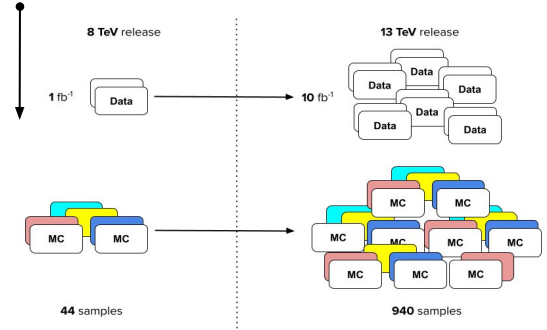
High-school community and general public
Some of the tools and apps target this audience to enhance particular projects like IPPOG-kind Masterclasses

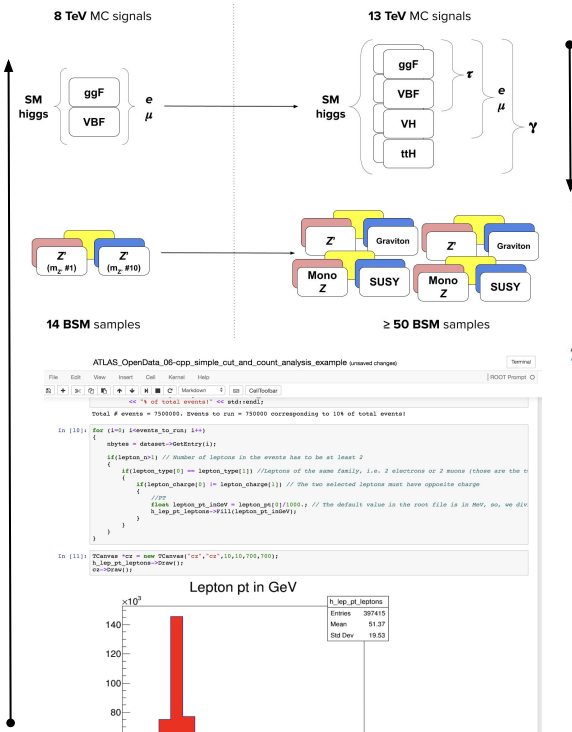
Those recipes and containers will be documented in a dedicated area of the opendata.atlas.cern site

The ATLAS Open Data project

In a nutshell, ATLAS OD resources are a series of

- Data samples in ROOT n-tuple
- Dedicated software & **Jupyter Notebooks** to produce physics analysis
- JavaScript applications to web-based produce cut-and-count analysis
- Virtual Machines with Jupyter, ROOT-CERN and other tools
- Series of **GitHub** & **GitLab** repos

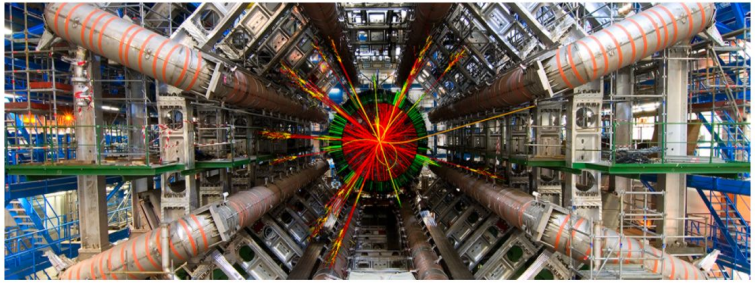




- Together with web-based documentation sites with resources and activities
- Multimedia tutorials (videos and exercises) complement the collection to be used by **students** (mainly in a lab or class)

The main entry point to the ATLAS Open Data project resources http://opendata.atlas.cern/

About



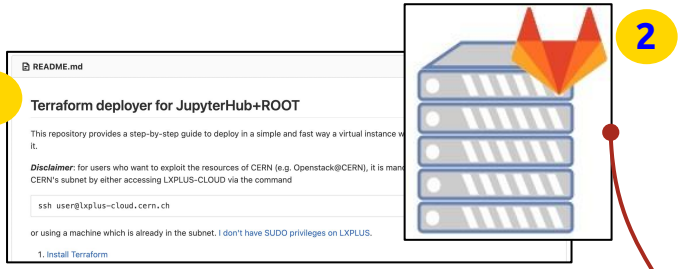
The ATLAS Open Data educational project

The aim of the ATLAS Open Data is to **provide data and tools** to high school, undergraduate and graduate students, as well as teachers and lecturers, to help educate and train them in analysis techniques used in experimental particle physics. Sharing data collected by the ATLAS experiment aims to generate excitement and enthusiasm for fundamental research, inspiring physicists of the future.

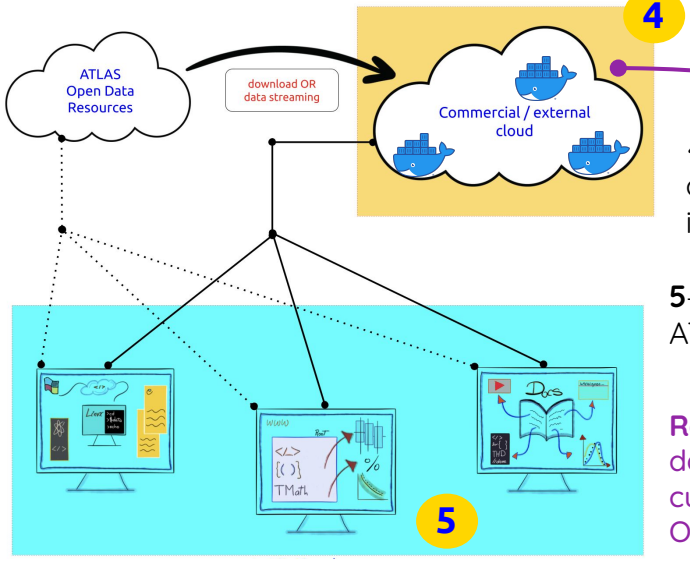
Current prototype: IaaS using Terraform

1 → A series of "recipes" for automatic deployment of instances (VM) in the cloud **2**→ The Terraform profiles+scripts are stored in GitLab, where some CI/CD is also done to ensure stability

3→ as with the containers, a dedicated page in the ATLAS OD website is the main entry point to discover how to use the solutions



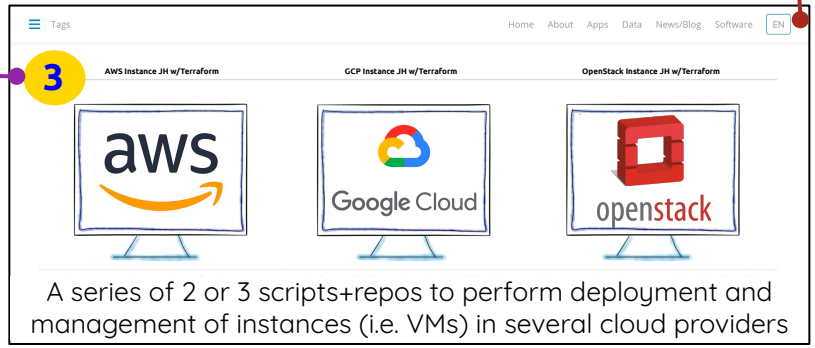
A multi-user architecture where the computers belong to an institutional infrastructure. The institution's cloud provides * SaaS, * the educational resources and datasets, * computing power, * permanent storage and * authentication methods.



IaaS using Terraform: the approach is the development of profiles that combine with scripts, allow the **automatic** creation, configuration and deployment of instances on several cloud computing providers.

4→ The single-user containers are spawned in the JupyterHub

5→ Users run and modified ATLAS OD examples



A series of 2 or 3 scripts+repos to perform deployment and management of instances (i.e. VMs) in several cloud providers

Result #1: After cloning the repo, the user needs ~5 min to execute a script and fill a few data (+ 10-15 min to get the Terraform CLI binary, no installation needed). After that, the current Terraform prototype delivers a working JupyterHub in a new VM in the CERN OpenStack in ~2 hours. Most of the time is dedicated to custom installations, like ROOT.

Some key references

- ATLAS Open Data: http://opendata.atlas.cern/
- ATLAS Outreach: on the dissemination of High Energy Physics and Computer Sciences. ATL-OREACH-PROC-2019-006: https://cds.cern.ch/record/2699514/
- CERN Open Data portal: https://opendata.cern.ch/
- JupyterHub https://jupyterhub.readthedocs.io/
- JupyterLab https://jupyterlab.readthedocs.io/
- Terraform: https://www.terraform.io/
- OpenStack: https://docs.openstack.org/
- SWAN Service at CERN: https://swan.web.cern.ch/
- Binder 2.0 - Reproducible, Interactive, Sharable Environments for Science at Scale. doi://10.25080/Majora-4af1f417-011

A concrete proposal: IaaS and SaaS

We are designing, creating, testing (and will release) a series of:

- **Terraform IaC recipes ready to use in cloud infrastructure**
  Current development uses the CERN OpenStack. Also, working on prototypes for AWS and GCP instances

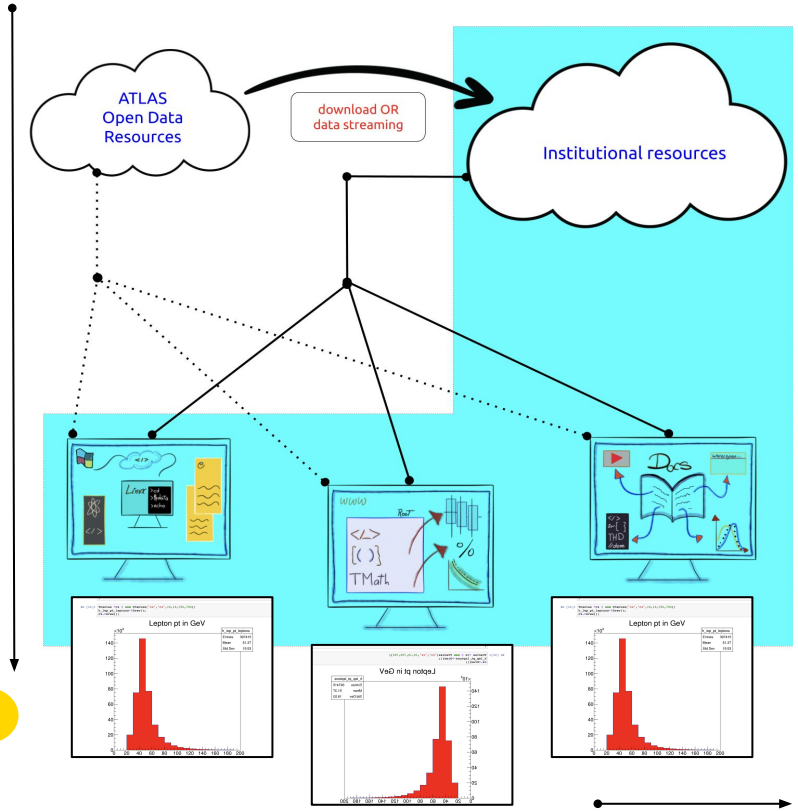
1

- **Containers that can replace or enhance the existing ATLAS OD single-user VM**
  Preliminary container solutions tested using DockerHub

2

Combinations of containers using JupyterHub Docker Spawner to deploy multiple-user JupyterHub with same single-user configuration

3



Users can deploy ATLAS OD resources in a cloud:

4

* On institutional resources out of the HEP community and with limited SysAdmin,
* or commercial clouds rented by institutions or individuals running short-term workshops

We explore the idea of:
Using the same single-user container in such multi-user environment, making possible to have a “single” product that fits multiple scenarios

5