

The logo graphic for ESCAPE features a stylized blue starburst at the top, a thin blue orbital line curving around it, and a small yellow circle at the bottom left of the orbit.

ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

Running an analysis on the ESCAPE VRE

A. Iess, CNRS LAPP



The ESCAPE Virtual Research Environment (VRE)

The **Virtual Research Environment** was developed by the VRE Team at CERN as part of the ESCAPE Project, under EU Horizon 2020 Grant Agreement no. 824064.

References

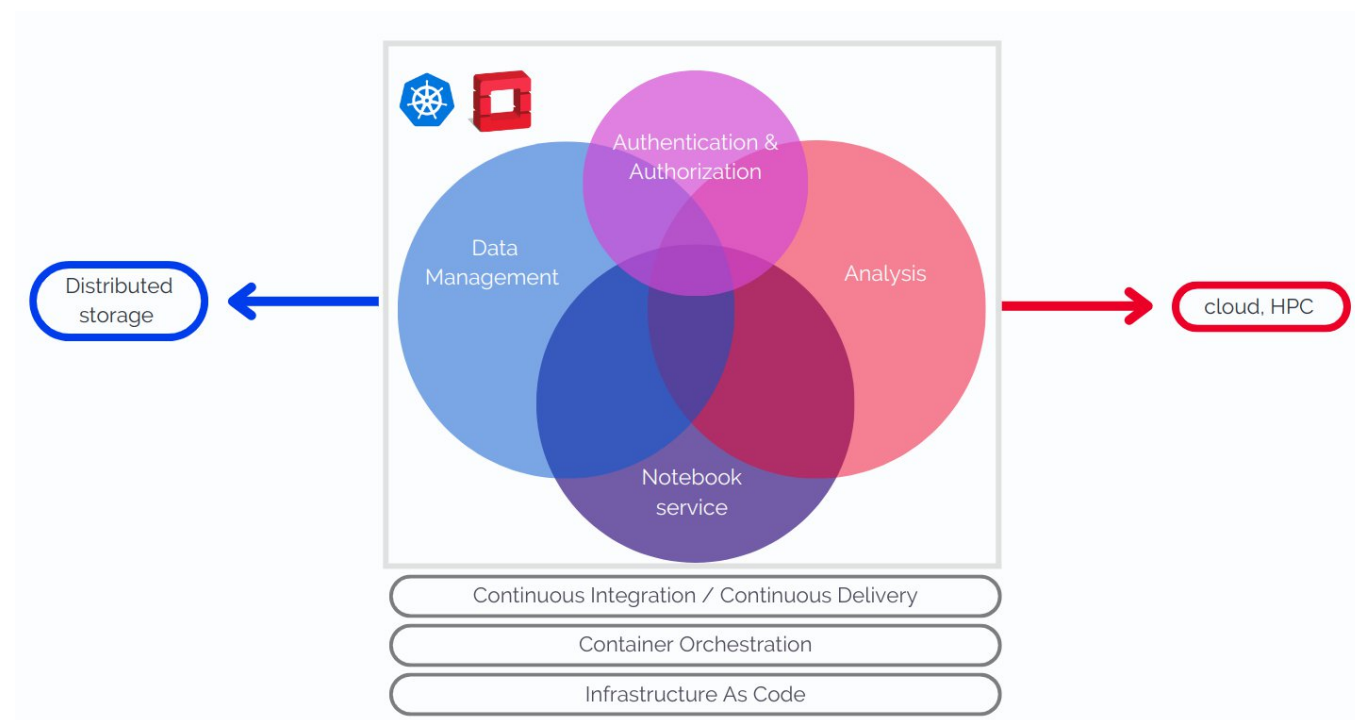
- CHEP2024 proceedings: <https://arxiv.org/pdf/2503.02483>.
- CHEP2024 contribution: <https://indico.cern.ch/event/1338689/contributions/6010696/>
- VRE Documentation: <https://vre-hub.github.io>
- Github Profile: <https://github.com/vre-hub>

CERN VRE Hub: <https://jhub-vre.cern.ch>

Mattermost Channel: <https://mattermost.web.cern.ch/escape/channels/vre-support>

VRE components

- Federated AAI
- ESCAPE Datalake for federated distributed storage .
- Computing cluster supplying the processing power to run full analyses.
- JupyterHub Interface with containerised environments.





ESCAPE AAI


- ESCAPE AAI is based on INDIGO Identity and Access Management (IAM).
- Request an account and wait for approval.
- Add escape group.
- Support Usr+pwd, JSON Web Token OIDC, x.509 certificates.

All ESCAPE Virtual Research Environment (VRE) services and resources are federated through the ESCAPE IAM service.



Welcome to **escape**

Sign in with your escape credentials

 Username

 Password

Sign in

[Forgot your password?](#)

Or sign in with

 Google

 eduGAIN

Not a member?

Apply for an account

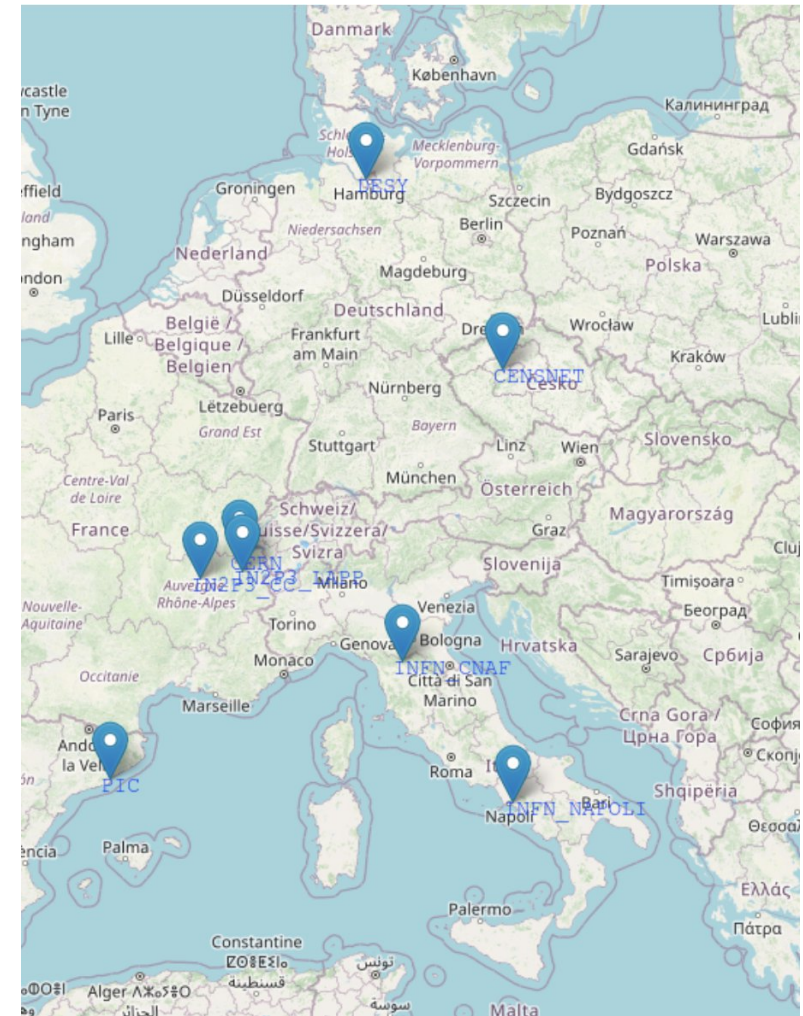
Register an account with eduGAIN

[Info and Privacy Policy](#)

You have been successfully authenticated as
**CN=IESS Alberto lkm3sm8x@cnrs.fr,O=Centre national de la
recherche scientifique,C=FR,DC=tcs,DC=terena,DC=org**
This certificate is not linked to any account in this organization

Rucio for Federated Distributed Data Management

- Distributed Rucio Storage Elements (RSEs).
- Rucio distributed data management system (upload, replication rules, transfer) .
- Interaction through docker container or rucio client.
- Authentication through X.509 certificates and OIDC tokens.



Running an analysis on the ESCAPE VRE

jupyterhub Home Token

The ESCAPE VRE offers a JupyterHub interface:

- AAI (credentials, x.509, OpenID)
- Environments encapsulated in Docker images and run as containers.
- Rucio, REANA plugins.
- VREs at [CERN](#) and at LAPP ([EOSC](#), [internal](#)).
- [Documentation](#).

You can select an environment and run a notebook interactively.

Server Options

- **Default environment**
Based on a scipy notebook environment with a python-3.11 kernel, the Rucio jupyterlab extension and the Reana client installed.
- **ROOT Higgs 2024 environment**
ROOT v6.32.04, and a python-3.11 kernel.
- **ROOT environment**
Legacy ROOT v6.26.10 as well as a ROOT C++ and a python-3.8 kernel.
- **VIRGO - WDF environment**
Contains the full WDF v2.2.3 environment and a Python 3.11 kernel.
- **Python 3.11 environment**
quay.io/jupyter/scipy-notebook:python-3.11 image
- **Default environment - python 3.9**
Same environment as the default one except for a python-3.9 kernel installed. This environment will be deprecated soon.
- **Default environment - python 3.8**
Same environment as the default one except for a python-3.8 kernel installed. This environment will be deprecated soon.
- **KM3Net Science Project environment**
Contains gammapy=1.1, km3lrf and km3net-testdata libraries - Python 3.9 kernel.
- **KM3NeT and CTA combined analysis environment**
Contains gammapy=0.17 and astropy - python 3.9 kernel.



Running an analysis on the ESCAPE VRE

Alternatively, add an environment with a pull request, which will be reviewed and eventually merged by the VRE team.

The screenshot shows the README for the `environments` repository. It includes a status bar at the top indicating "Docker automatic build and publish" is passing. The main heading is "Environments", followed by a description: "VRE user environment images for workflows and notebooks available in the VRE JupyterHub service." Below this, there are sections for "VRE user environments" and "Custom user environments". The "Custom user environments" section provides instructions on how to extend or modify an image, recommending the use of the latest `vre-singleuser-py311` image as a base layer. A list of suggestions includes checking the latest version of the base image, customizing it, creating a folder per environment, and adding a `README.md` file. At the bottom, a code block shows the beginning of a suggested Dockerfile:

```
# Beginning of the suggested Dockerfile
FROM ghcr.io/vre-hub/vre-singleuser-py311:sha-5106e39
LABEL maintainer=<your_name>
ARG BUILD_DATE
LABEL org.label-schema.build-date=$BUILD_DATE
...
```

The screenshot shows the GitHub repository page for `environments`. The repository is public and has 16 branches and 0 tags. The main branch is `main`. The repository contains a list of environments, each with a description and a commit hash. The environments listed are:

- `GravitationalWaveSurfer`: Feat: add WDF python 3.11 env (#60) - 15 hours ago - 66 Commits
- `.github`: do not trigger second stage, what makes CI fail, if the o... - 6 months ago
- `atlas-ntuples`: change name of repo - 2 years ago
- `vre-singleuser-combined_ana_km3net_cta`: some changes in readme for cta env (#32) - 2 years ago
- `vre-singleuser-dask-root`: adding image with rucio extension and dask (#41) - 2 years ago
- `vre-singleuser-interTwin`: VRE - interTwin DL integration (#45) - 8 months ago
- `vre-singleuser-itwinai`: Itwinai - Rucio jlab extension integration (#50) - 5 months ago
- `vre-singleuser-km3irf`: Km3irf env (#34) - 2 years ago
- `vre-singleuser-microomega`: WIP: migrate microomega docker file to gh and test buil... - 2 years ago
- `vre-singleuser-py311`: Copy correctly vomsdir content (#56) - 3 months ago
- `vre-singleuser-py38`: update py38 base image in agreement wiht latests chan... - 2 years ago
- `vre-singleuser-root`: Include rsync in packages (#58) - 2 months ago
- `vre-singleuser-wdf`: Feat: add WDF python 3.11 env (#60) - 15 hours ago
- `vre-singleuser`: Trigger build of singleuser (#37) - 2 years ago
- `AUTHORS.md`: some changes in readme for cta env (#32) - 2 years ago
- `LICENSE`: Update LICENSE - 2 years ago
- `README.md`: update environment information - 6 months ago

The right sidebar shows repository statistics: 2 stars, 4 watching, 1 fork, and 0 releases. It also lists packages (25) and contributors (6).

Running an analysis on the ESCAPE VRE

RUCIO

EXPLORE NOTEBOOK

ET_OSB_MDC1v2:*

Search Everything

SEARCH RESULTS

Folder	Size
ET_OSB_MDC1v2:GW.Frames.DataSim	
ET_OSB_MDC1v2:GW.FramesE0.DataSim	
ET_OSB_MDC1v2:GW.FramesE1.DataSim	
ET_OSB_MDC1v2:GW.FramesE2.DataSim	
ET_OSB_MDC1v2:GW.FramesE3.DataSim	
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000000000-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000002048-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000004096-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000006144-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000008192-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000010240-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000012288-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000014336-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000016384-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E0_STRAIN_DATA-1000018432-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E1_STRAIN_DATA-1000000000-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E1_STRAIN_DATA-1000002048-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E1_STRAIN_DATA-1000004096-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E1_STRAIN_DATA-1000006144-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E1_STRAIN_DATA-1000008192-2048.gwf	122.71MiB
ET_OSB_MDC1v2:E-E1_STRAIN_DATA-1000010240-2048.gwf	122.71MiB

Untitled.ipynb | Terminal 2

Python 3 (ipykernel)

```
[2]: # import libraries
import time
import os
from pytsa.tsa import *
from pytsa.tsa import SeqView_double_t as SV
from wdf.config.Parameters import *
from wdf.processes.wdfUnitDSWorker import *
from wdf.processes.wdfUnitWorker import *
import logging
import coloredlogs
#select level of logging
coloredlogs.install(isatty=True)

logging.basicConfig(level=logging.DEBUG)
```

Rucio Data Lake Extension

```
[1]: ew_json_config_file = True # set to True if you want to create new Configuration

if new_json_config_file==True:
    configuration = {
        "window":1024,
        "overlap":768,
        "threshold": 0.2,
        "file": "./data/test.gwf",
        "channel": "H1:GWOSC-4KHZ_R1_STRAIN",
        "run":"offLine",
        "len":10.0,
        "gps":1167559608,
        "segments":[[[1167559608,1167559408],[1167559408,1167560008],[1167560008,1167560308]],
        #segments:[[1167559608,1167560008] ],
        "outdir": "local_dir/",
        "dir":"local_dir/",
        "ID":"WDF_test",
        "ARorder": 1000,
        "learn": 200,
        "preWhite":2,
        "ResamplingFactor":2,
        "LowFrequencyCut":12,
        "FilterOrder":6,
        "nproc":4
    }

    filejson = os.path.join(os.getcwd(),"inputWDF.json")
    file_json = open(filejson, "w+")
    json.dump(configuration, file_json)
    file_json.close()

logging.info("read parameters from JSON file")
par = Parameters()

filejson = "inputWDF.json"
try:
    par.load(filejson)
except IOError:
    logging.error("Cannot find resource file " + filejson)
    quit()

par.print()
```


Reproducible analysis platform for containerised data analysis pipelines on remote compute clouds.

- Supported workflow systems:
[CWLSerial](#), [Snakemake](#), [Yadage](#)
- Supported compute backends:
[HTCondor](#), [Kubernetes](#), [Slurm](#)
- Supported source code and storage systems: GitLab, CVMFS, EOS

