



ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures

Lessons learned from integrating ESAP with EGI Cloud Computing

S. Bertocco

ESCAPE ESAP Training Workshop

21-22 November 2022



ESCAPE: European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures.

The project aims to address the Open Science challenges shared by ESFRI facilities (SKA, CTA, KM3Net, EST, ELT, HL-LHC, FAIR) and pan-European research infrastructures (CERN, ESO, JIVE) in astronomy and particle physics.

Connect ESFRI projects to EOSC ensuring integration of data and tools

Establish interoperability within EOSC as an integrated multi-messenger facility for fundamental science.



ESAP: ESFRI Science Analysis Platform.

Flexible science platform for the analysis of open access data available through the EOSC environment. It allows EOSC researchers

- ▶ to identify and stage existing data collections for analysis,
- ▶ to select among software tools and packages developed by the ESFRIs or bring their own custom workflows to the platform,
- ▶ to exploit the underlying computing infrastructure to execute those workflows.



Two use cases:

- ▶ provide IVOA services in a container

<https://github.com/zarquan/Oligia>

- ▶ provide an environment to run the workflow to process HI data cubes produced by radio interferometers, in particular large data cubes produced by future instruments like the SKA.

<https://hi-friends-sdc2.readthedocs.io/en/latest/>

<https://github.com/Hi-FRIENDS-SDC2/hi-friends>

which is a suitable infrastructure to satisfy such use cases ?



EGI Fedcloud in a nutshell

EGI is a federation of computing and storage resource providers delivering open solutions for advanced computing and data analytics to support research and development

The EGI Core is a federation and management platform that pool together various resources: the EGI Federated Cloud Platform (FedCloud)

The EGI cloud compute service gives users the ability to deploy and scale virtual machines on-demand.



- **Cloud Compute**: VM-based computing with associated storage.
Typical use-cases are user gateways or portals, interactive computing platforms and almost any kind of data- and/or compute-intensive workloads.
- **Container Compute**: supports running container-based applications with either Docker or Kubernetes on top of Cloud Compute. Typical use-cases are multi-tenant, microservices-based applications that must easily scale horizontally.
- **High Throughput Compute** provides access to large, shared grid computing systems for running computational jobs at scale.

<https://docs.egi.eu/users/compute/>



Access EGI Resources - Instructions

- Signing up
- Create an EGI account with Check-in.
- Enrolling to a Virtual Organisation
- vo.access.egi.eu is a dedicated VO for piloting purposes
- Creating a VM
- Ways to achieve this:
 - web dashboard
 - command-line interface (FedCloud client)

<https://docs.egi.eu/users/tutorials/create-your-first-virtual-machine/>
<https://github.com/tdviet/fedcloudclient>



Implementation steps

- ▶ register to the Virtual Organisation vo.access.egi.eu
- ▶ ask to CESGA site support to be able to allocate needed resources (particularly a public IP)
- ▶ write a simple django web application, to be integrated in ESAP, to automate the virtual machine creation process in the EGI fedcloud
 - following the EGI provided guide
<https://docs.egi.eu/users/getting-started>
 - using the EGI federated cloud python client
<https://github.com/tdviet/fedcloudclient>



Implementation steps

EGI Federated Cloud Resources Access API

Home

- Create Server
- Delete Server
- Show Server
- Create Access Key

Here are available the EGI Federated Cloud Resources Access utilities. Before to create a server you must do three things:

- Create your ssh key-pair following the instructions [here](#)
- Load your public key using the "Create Access Key" in the menu on the left
- Create you own EGI fedcloud access token following the instructions [here](#)

If needed, instructions to ask resources are [here](#)

Create Server

Home

- Create Server
- Delete Server
- Show Server
- Create Access Key

Access Token*:

Provide a server name*:

Pre-loaded ssh key name*:

Server creation result

Home

- Create Server
- Delete Server
- Show Server
- Create Access Key

Server successfully created:

Server name: ESCAPE TEST
 Server IP: XXX.XXX.XX.XXX
 Access user: ubuntu
 Connect using: ssh ubuntu@ XXX.XXX.XX.XXX

<https://github.com/bertocco/ESCAPE-VM-worker>
<https://github.com/bertocco/ESCAPE-VM-worker-gui>



Lessons learned

Show stopper in the virtual machine automation process:

1) Public network. Each site has a slightly different network configuration, and there is no standard way to tell which one to use and how to setup the router for it. There is not a convention for the Networks naming → need some guess (human intervention) to understand which is a public network.

2) Supported Virtual Organizations. There is not a command in openstack or in the Fedcloud tool that lists the sites that support a certain virtual organisation.

3) Flavours and images: it is needed to guess (human intervention) the resources associated to the flavors or the Operating System in an image. A naming convention could be useful. Possible solution for flavours: to use 'list' and 'show' commands in association to create a configuration information file.



Point 3 workaround proposal

Collect informations:

openstack image list

list available flavors

openstack image show <image>

display flavor details

openstack flavor list

list available flavors

openstack flavor show <flavor>

display flavor details

to create a configuration information service to query to get the needed information



Authentication and Authorization thoughts

- ▶ ESCAPE-ESAP **A&A** managed with **INDIGO-IAM**

<https://indigo-iam.github.io/v/current/docs/>

Project deployment:

<https://iam-escape.cloud.cnaf.infn.it/login>

- ▶ EGI FedCloud **A&A** managed with **EGI Check-in**

<https://docs.egi.eu/users/aai/check-in/>

EGI Check-in as a service:

<https://aai.egi.eu/registry/>

Both: OpenID Connect, JWT token, group membership

Could them interoperate?

Authentication: mapping between tokens?

Authorization: based on group membership



VM access using tokens ?

Future plan: evaluating **motley cue**

(Mapper Oidc To Local idEntitY with IoCal User managEment)

A service for mapping OIDC identities to local identities.

<https://motley-cue.readthedocs.io/en/latest/>



Conclusions

- ▶ Software VM creation in EGI fedcloud is feasible
- ▶ Automation is challenging
 - Work needed to standardize information to make possible automation
- ▶ Work is needed to make homogeneous authentication and authorization
- ▶ Future plan: Evaluate and try EGI container compute



Thanks to

Susana Sanchez Exposito

Dave Morris

Manuel Parra-Royón

Klaas Kliffen

John Swinbank

Enol Fernández (EGI)





ESCAPE

European Science Cluster of Astronomy &
Particle physics ESFRI research Infrastructures



Thank you !

