

**EGI-Engage**

Assessing the EGI Federated Cloud and  
its Possible Applications for Dutch Science

|  |  |
| --- | --- |
| **Date** | 07 November 2017 |
| **Activity** | SA2, ELIXIR Competence Centre |
| **Authors** | Natalie Danezi, Markus van Dijk, Nuno Ferreira (SURFsara) |
| **Document Status** | Final |
| **Document Link** | https://documents.egi.eu/document/XXX |

Abstract

This document describes the EGI Federated Cloud as seen from the national perspective of SURFsara, the institute that represents the Dutch national e-infrastructure in EGI. It highlights the main features of the infrastructure and describes its functionality. The document goes into the value of the federated cloud to Dutch science and evaluates the effort that is needed to connect national resource to this European initiative. The document was produced in 2017 the context of the ELIXIR Competence Centre of the EGI-Engage H2020 project.

**COPYRIGHT NOTICE**



This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/). The EGI-Engage project is co-funded by the European Union Horizon 2020 programme under grant number 654142.

**DISCLAIMER**

The views, thoughts, and statements expressed in the document are of the authors and do not necessarily reflect the official position of the employer (SURFsara) or the project context in which this document was published.

**TERMINOLOGY**

A complete project glossary and acronyms are provided at the following pages:

* <https://wiki.egi.eu/wiki/Glossary>
* <https://wiki.egi.eu/wiki/Acronyms>

**Contents**

1 Introduction 5

2 The EGI Federated Cloud 6

2.1 The architecture 6

2.2 Users 7

2.2.1 User profile 7

2.2.2 User access 8

2.2.3 Future developments 10

2.3 Access channels explained 10

2.3.1 AppDB 10

2.3.2 VMOps portal 11

2.3.3 Command line and APIs 14

2.3.4 Applications on Demand 16

2.4 Resource providers 20

2.5 Authentication and authorisation 22

2.6 Data management 23

3 Evaluation of support and usability 25

3.1 Documentation 25

3.2 Enforcement 25

3.2.1 Accounting and billing 25

3.3 SLA, SW ownership 26

4 Conclusions 27

4.1 Overall 27

4.2 Considerations for adoption 28

4.2.1 Technical implications 28

4.2.2 Value for user community 30

5 References 32

**Executive summary**

The EGI Federated Cloud has improved considerably since the initial evaluation in 2014. Many of the problems encountered then have now been fixed and the platform itself is more mature and better supported, most notably in offered services, documentation and accounting. It has reached a level where novices to federated e-infrastructure can use the provided resources with relative ease.

There are, however, a number of issues still to be addressed. To start with, the long-term vision on the Federated Cloud needs to be made clear: will it work towards a broker role, distributing VMs across sites, or will it focus on federation, integrating different cloud providers in such a way that they are hidden from the end-user? Standardisation of federated service management also needs to be worked on, the flexibility of individual SLAs between service providers and consumers is appreciated but might benefit from pre-defined service levels.

There are also a number of technical issues that need to be looked into. For one, data distribution and the way cross datacentre data distribution and look up need to be performed. Another is AAI, where multiple bugs have been discovered in the course of this project.

Adoption of Federated Cloud Services and offering part of the Dutch National Infrastructure through this channel relies on decision of upper management. From a practical point of view, joining the Federated Cloud with one of the Dutch OpenStack instances will, even given the expertise gained by operating grid services, be resource intensive. The added workload caused by added maintenance overhead and additional tickets will not be trivial.

The added value of the Federated Cloud comes from two factors: (1) providing services not found on a national level, and (2) providing services in an international context. For both of these the case can be made that researchers in the Netherlands would benefit from integration of the DNI services with the EGI Federated Cloud.

# Introduction

[SURFsara](https://www.surf.nl/en/about-surf/subsidiaries/surfsara/) is an [EGI](https://www.egi.eu/) member that delivers advanced computing and storage services to researchers in the Netherlands. Thus, we realise first hand that the researchers wish to have easy access to IT resources where they can select predefined workflows, applications, compute and data storage. At the same time background processes and requirements like accounting, security and privacy should be provisioned where possible.

Hence, SURFsara is investigating scalable services on platforms that seamlessly federate, integrate and provide access to distributed services, tools and FAIR data in order to support research on national and European level. This is in line with the general vision of the European Open Science Cloud (EOSC).

With all that, our interest in the EGI Federated Cloud becomes apparent. EGI has been offering Grid Computing services to European researchers and their international partners for over a decade. Since 2014, EGI has put into production Cloud computing services too, forming the EGI Federated Cloud that integrates 24 sites across Europe. Therefore, we evaluate here the usability and feasibility of the EGI Federated Cloud services and how the Dutch user communities could benefit from it.

With this document we introduce the capabilities and limitations of the EGI Fed Cloud internally to SURFsara to help also our management assessing the possibility of SURFsara participation in the federation as a resource provider. In addition, the evaluation will help us learn from the implementation technologies chosen by EGI as we share some common services in our catalogue.

Finally, knowing the developments in EGI Federated Cloud is important for our positioning in the EU science spectrum as we are involved in several international initiatives like EGI, PRACE, EOSChub, Elixir, HelixNebula, Cern, Lofar and WeNMR.

# The EGI Federated Cloud

EGI is offering services to European researchers and their international partners. The EGI Federated Cloud service catalogue has been recently updated to a much clearer model that allows researchers and partners easily search and apply for the supported services. The clarity of EGI service catalogue is also reflected on their easy-to-navigate website.

Taken from the [EGI website](https://www.egi.eu/services/), a quick summary of the services is given in this table:

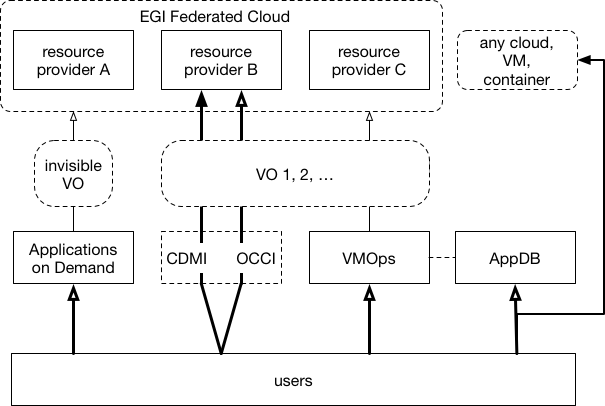
|  |  |
| --- | --- |
| **Name** | **Description** |
| Compute | Cloud Compute, Cloud Container Compute *(beta)*, High-Throughput Compute *(Grid)* |
| Storage and Data | Online Storage, Archive, Data Transfer *(FTS)* |
| Applications | Applications on Demand *(beta)*, scientific applications *(rely on EGI services)* |
| Training | FitSM, training infrastructure |

Here, we focus on the Cloud-based services dedicated to scientific processing which are described below:

* *Cloud Compute* service: enables running virtual machines on-demand with complete control over computing resources.
* *Cloud Container Compute* service: enables running Docker containers in a lightweight virtualised environment.
* *Online Storage* service: enables storing, sharing and accessing files and their metadata on a global scale.
* *Applications on Demand* service: enables deploying platforms such as Jupyter Notebooks to support scientific processing.

## The architecture

The EGI Federated Cloud sites and their interconnections are based on open standards and open technologies. Users can make use of the Cloud-based services provided by various Cloud providers via different channels depending on their applications and expertise. In the following diagram we summarise the main blocks of EGI Federated Cloud in a graphical overview. The diagram is built from the user perspective:



Below we explain the core elements and roles forming the different layers in the architecture: users, access channels and resource providers. We also describe the authentication and data management models used in the federation to include all the main components of a project lifecycle.

## Users

### User profile

The EGI Federated Cloud is based on a user-centric model. We identified the following user groups that can take benefit from the EGI Federated Cloud services:

* individual scientists
* members of established scientific communities
* application developers and service providers of established scientific communities

The individual scientists or 'long tail' of scientists is a large and growing group of researchers who do not require highly advanced, specific solutions. Their profile is a novice user that has limited experience or interest in IT, thus requires easy access to compute, storage and network resources.

The members of an established scientific community or 'Virtual Organisation (VO)' are scientists with the same profile as the first group but having also access to tools and support provided by the service providers in their community. They typically require tools that hide the infrastructure complexity and help collaboration in analysis of multiple and distributed data sources.

The application developers and service providers of scientific communities is a group of people who develop tools, design community-oriented services and define policies targeted to the previous group. They aim to create a 'virtual workspace' with integral interface to all data, infrastructure and analysis tools. They also take care that their services are compliant with privacy laws and meet a certain level of security and privacy.

Examples of scientific communities active or interested in EGI Federated Cloud services are: Elixir (Bioinformatics), DARIAH (Arts and humanities), MoBrain (Structural biology), BBMRI (Biobanking), LifeWatch (Biodiversity sciences), EISCAT\_3D (Ionosphere and atmosphere observatory) and EPOS (Earth sciences).

The common ground between all three user groups is the need for flexibility and elasticity. They all need a flexible environment to deploy controlled applications that can scale rapidly to extended e-Infrastructure across resource providers in Europe in order to cope with their demand.

### User access

The user groups can make use of the EGI Federated Cloud services by choosing one of the following main access channels (see diagram):

* AppDB Cloud Marketplace
* VMOps Dashboard
* Command-line open standards & APIs
* Applications on Demand

The users can deploy new appliances (images and templates) to a Cloud Marketplace, called AppDB. The AppDB appliances can be instantiated on any EGI Federated Cloud site that supports a particular VO. Some of the AppDB images can be transformed and used on an another Cloud that is not part of the federation, e.g. the HPC Cloud. The preparation and deployment of an appliance requires understanding of several Cloud technologies like creating and packaging images. Thus, we consider that this task should be handled by application developers or service providers managing the VO, rather than individual scientists. On the other hand, browsing and exploring the available appliances on the AppDB is quite straightforward for any scientist due to the web-based graphical environment and the various filtering options to discover the appliances.

Once the user selects an appliance from the AppDB Cloud Marketplace, he has two main options to spawn the corresponding VM on the EGI Federated Cloud sites. One option is to instantiate the VM in few 'clicks' from the VMOps dashboard that provides a graphical interface. Another option is to start a VM directly via the command-line clients or the supported APIs. We find that the command-line interfaces and API scripting are too complicated for a novice user, but necessary for the application developers and service providers. For the scientists we would recommend the VMOps Dashboard that has a clean and simple user interface.

Users who wish to deploy an application on the EGI Federated Cloud without knowing the details of the appliances, can make use of high level tools, such as the Applications on Demand (AoD) service. The Applications on Demand gives access to services like Chipster, Galaxy, Docker and Jupyter Notebooks. The users are given pre-defined quota of resources, which can be used to run an application of their choice. We find this service well designed to provision VMs on the PaaS and SaaS level.

Currently the AppDB Cloud Marketplace is publicly available to anyone, while the use of VMOps Dashboard, command line interfaces and APIs require a VO membership and thus, a Grid certificate. Granting access to the AoD service does not require a VO membership nor a Grid certificate, nor advanced experience with distributed and/or cloud computing. The goal is that all of these services will be soon integrated with an authentication model that does not depend on Grid certificate authentication.

### Future developments

A new service, called Marketplace, will soon be released to improve the discovery of the EGI Federated Cloud services. In the Marketplace portal, the users would be able to order services such as VMs, Jupyter Notebooks or block storage. Then, users can check out the selected items via their EGICheckIn account and start deploying the granted services on the VMops and AoD platforms. The design of the Marketplace is well-defined, includes functionalities tailored to users and also displays information related to the grant such as terms of use, SLAs and access policies.

## Access channels explained

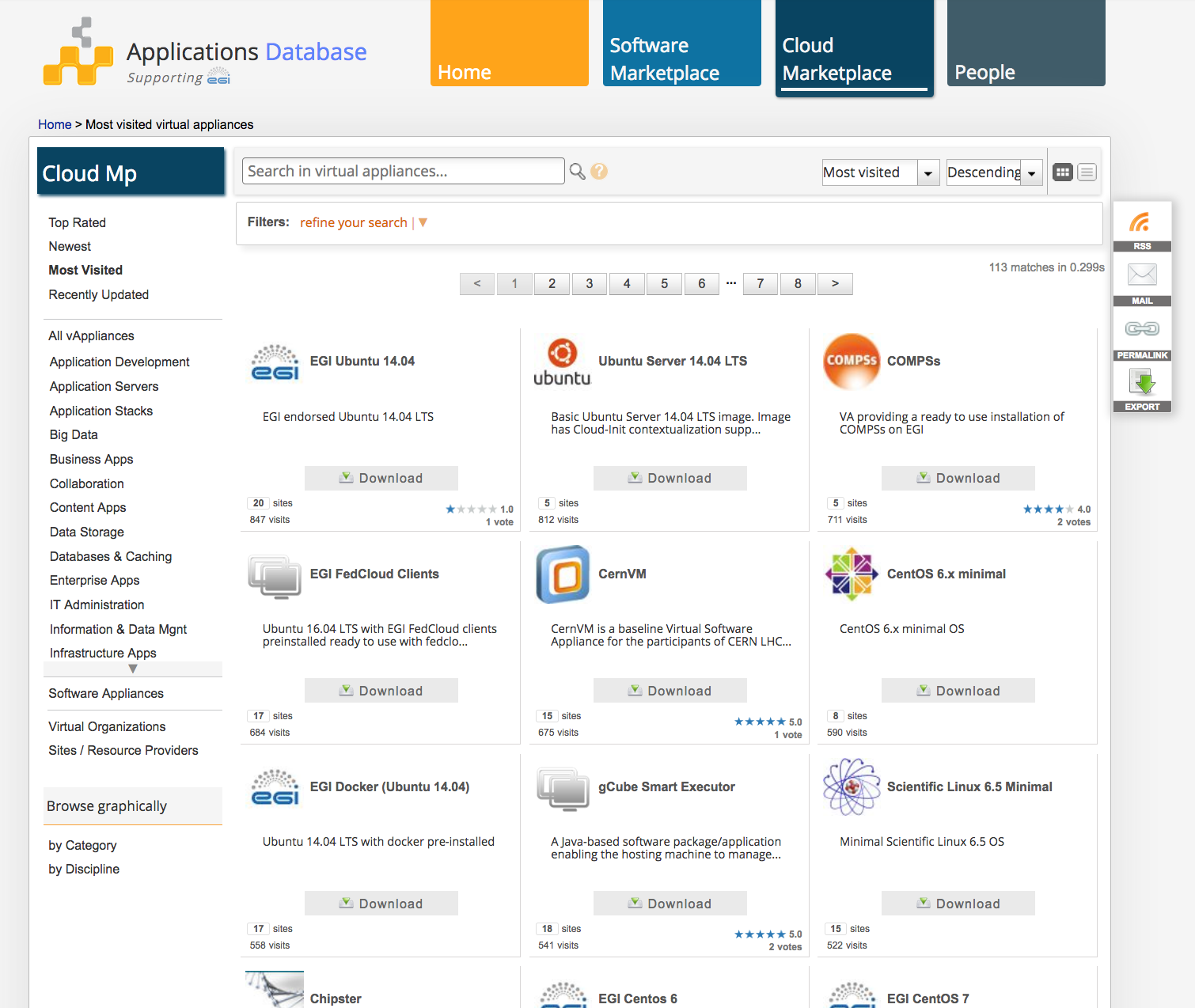
### AppDB

The EGI Application DataBase (AppDB) is a central service that stores and provides information about software products, virtual appliances and software appliances. The AppDB has a Software Marketplace and a Cloud Marketplace. The AppDB Cloud Marketplace offers the possibility to select pre-configured virtual appliances (e.g. CPU, memory, disk, operating system and software). The appliance is stored in a catalogue that is replicated across the EGI cloud providers.

The main functionalities that the [AppDB Cloud Marketplace](https://appdb.egi.eu/browse/cloud) offers are:

* Searching among existing virtual appliances and their descriptions.
* Discovering which EGI sites support a given appliance.
* Retrieving the appliance IDs which is a combination of the OS template ID (image) and the Resource template ID (VM specifications).
* Filtering the appliances supported at each EGI site.
* Filtering the available appliances per VO and the EGI sites supporting them. For example, at the time of writing the Elixir VO vo.elixir-europe.org provides appliances for basic Linux distributions, Chipster and Docker which are supported on 3 Sites: CESNET, GENET, IN2P3.
* Sharing and distributing customised VM images. Community curated image VMs are securely and automatically replicated across the infrastructure.
* Downloading a (trusted) appliance for your own deployment on other Clouds.

The AppDB has a graphical interface and a RESTful API. Here is a view of the web-based interface:



The base URI of the RESTful API is http://appdb-pi.egi.eu/rest. API requests must be followed by at least a resource name e.g. /applications, /va\_providers. There are java and python clients available to query the API.

The EGI AppDB has been recently integrated with the VMops portal, allowing authorised users to perform some basic VM management operations.

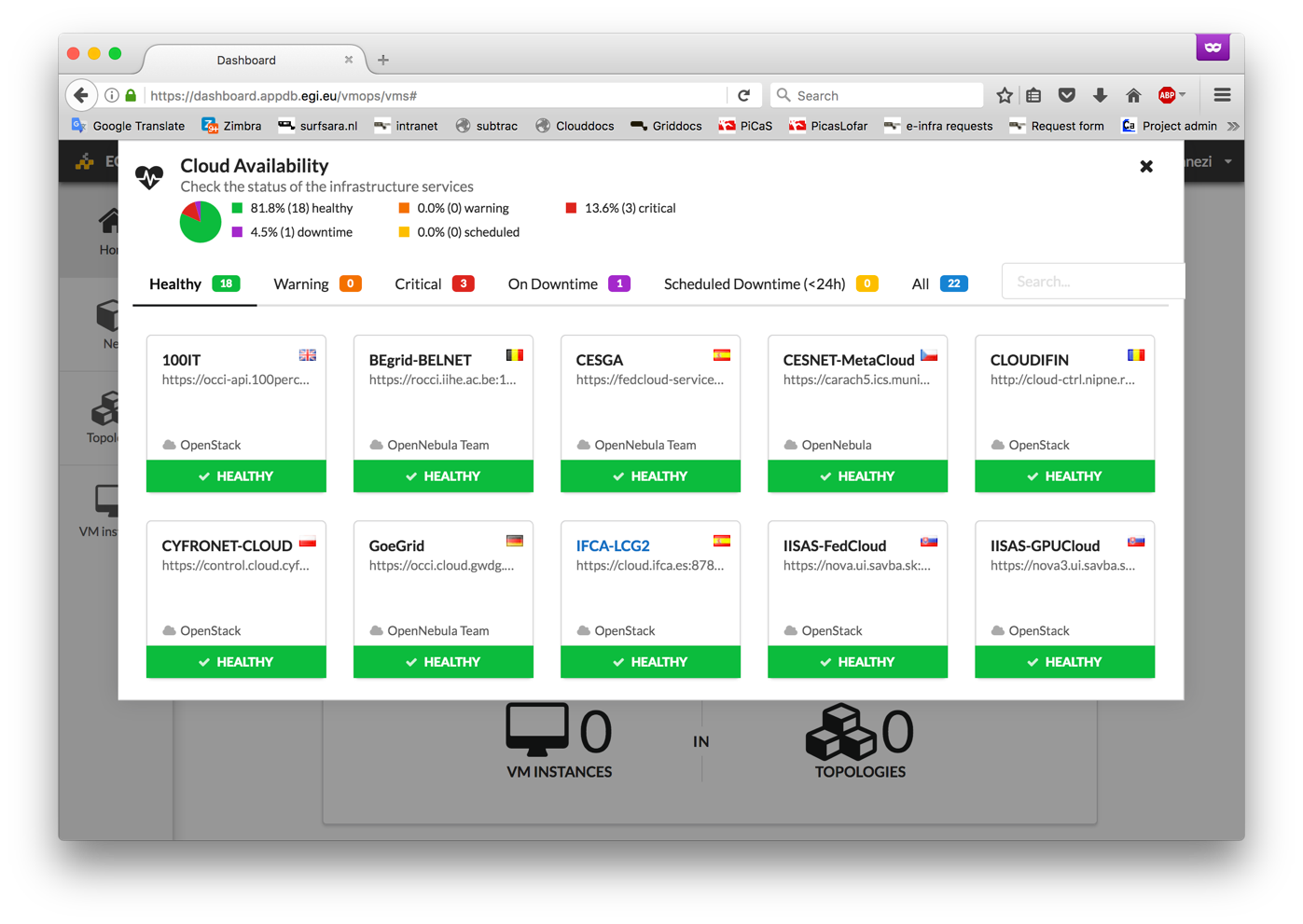
The EGI AppDB GUI can be accessed either anonymously or via the VO membership and Grid proxy credentials.

### VMOps portal

The Virtual Machine Operations ([VMOps](https://dashboard.appdb.egi.eu/vmops)) provides a Graphical User Interface (GUI) where users can create and manage their VMs on the EGI Federated Cloud with a 'wizard-like' topology builder. This is a very useful addition to the existing services as it simplifies the VM instantiation process a lot. The VMops dashboard hides all the Grid toolkit dependencies that were previously exposed in the command-line interfaces.

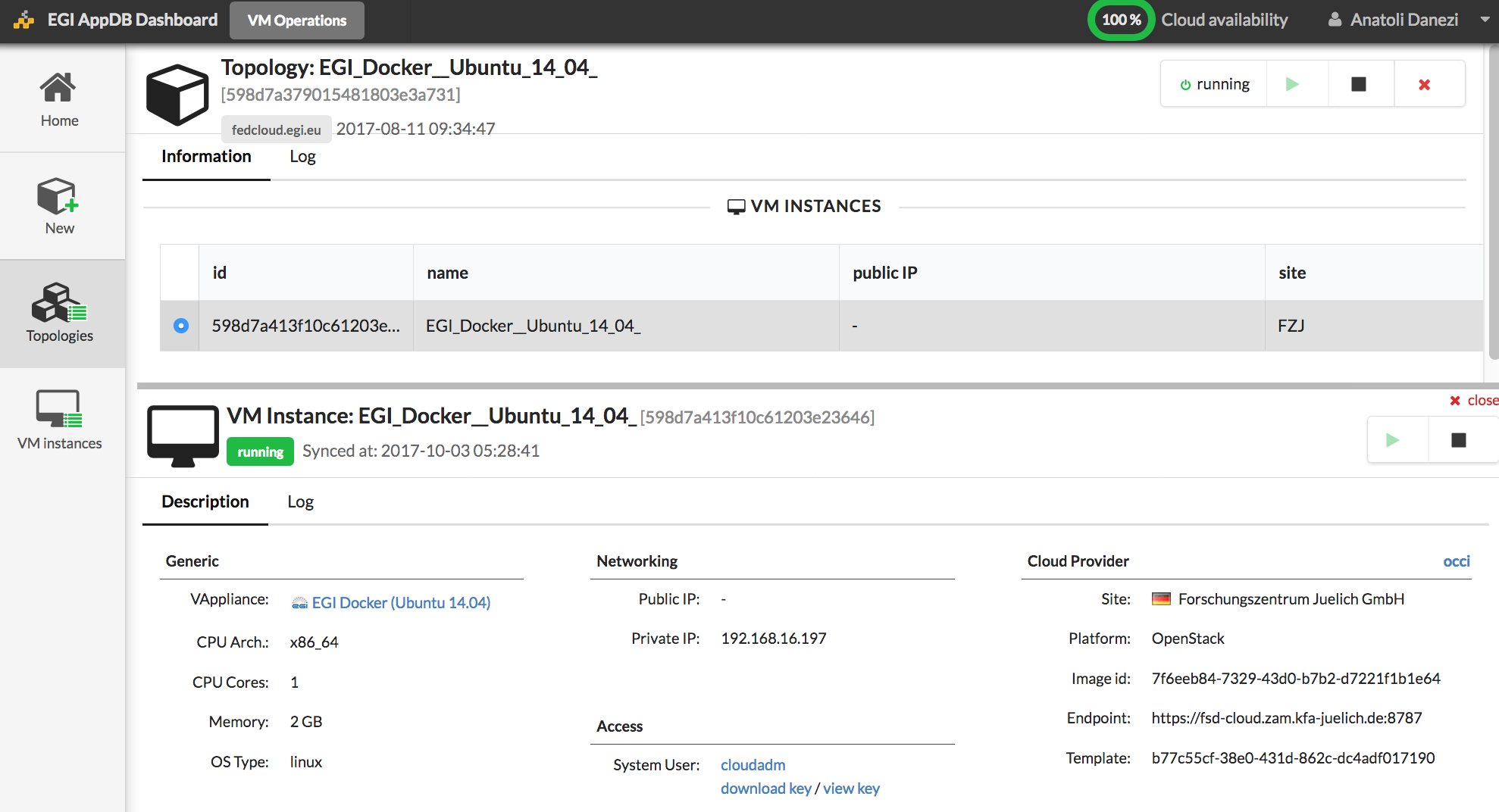
The VMOps portal allows the basic VM management functionalities listed here:

* Creating and managing topologies of VMs.
* Starting/stopping individual VM instances within a topology.
* Adding block storage devices to the VMs.
* Displaying the healthiness of all federated sites. For providers in warning, critical or downtime state there is a link to the Configuration Database (GOCDB) (see picture below).
* Linking the available appliances to AppDB where user can find information on the images supported per site.



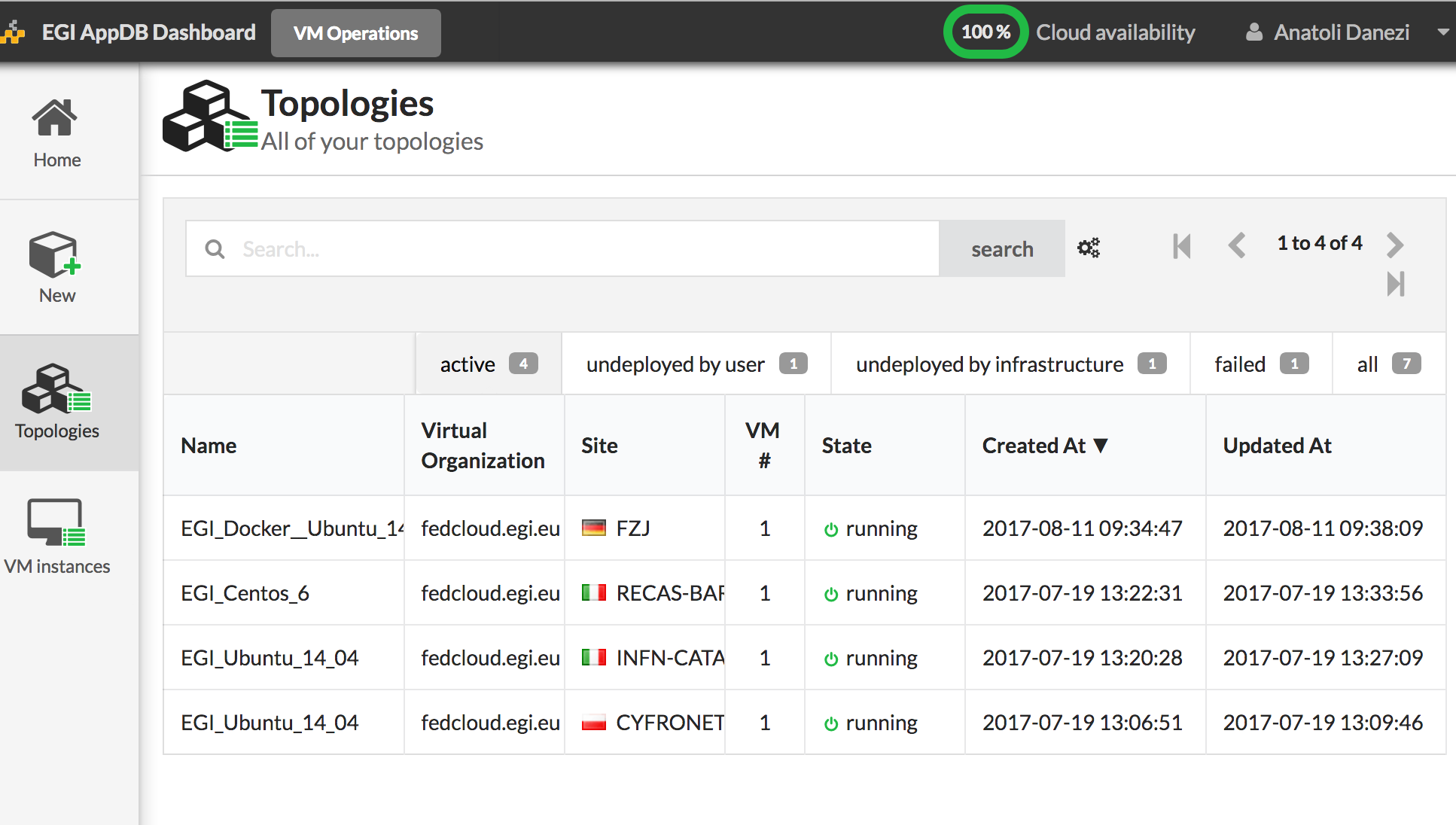
Once a VM is created on the VMops portal, the user can connect to it with an ssh client by using the VM public IP address and his public key. The public key is injected in the machine with a contextualization script. EGI Federated Cloud VMs are using the Cloud-init tool for contextualization to configure network, users, ssh keys, filesystems, install packages, execute arbitrary commands and execute user provided scripts inside the VM.

Clicking on a VM of a topology on the VMops portal, the details about its status and IP are shown. Even if a user or ssh key is not specified, the AppDB creates one and provides the credentials for login on the dashboard, see picture:



The VMops is OCCI-based and interacts with the site providers via the TOSCA standard. It is designed with a scalable architecture composed by a front-end and several back-ends for load balancing and it has a RESTful API.

Most important, there are only 3 buttons available to the user: a) play b) stop (suspend topology) c) delete (undeploy topology) which improves the user experience a lot, compared to the native GUIs offered by OpenNebula and OpenStack. The picture below depicts the VMops dashboard:



The EGI AppDB can be accessed either anonymously or via the VO membership and Grid proxy credentials. It is possible to connect an account to others like SSO or eduGain. At the time of writing, the VMops topology builder is usable only by using the Grid proxy installed in the browser and a VO membership in fedcloud.egi.eu, but it is working progress adding other EGI checkin authentication methods that don't require Grid certificates. Removing the need for users to own X.509 certificates and moreover, the complete EGICheckIn integration is one of the VMops implementation goals.

### Command line and APIs

Except for the VMops portal, the users can create and manage their VMs' resources via command-line interfaces (CLI) and APIs. This was the initial access method available at the start of EGI Federated Cloud which has been extensively presented in the previous EGI Federated Cloud evaluation, in 2014. Except for some new 'best practices' documented on the EGI user wiki, there are no remarkable changes, so we will only make here a quick summary of CLI functionalities.

To start with the command line clients, a user needs to setup the proper environment. For setting up the EGI Federated Cloud command-line environment, we have tried different options as instructed on the EGI wiki:

* Deploying a VirtualBox UI image on our laptop.
* Deploying the AppDB UI image on the HPC Cloud. This required conversion from OVA to RAW format.
* Executing installation scripts on bare-metal.
* Starting the UI in a Docker container on our laptop.

It proved that the VirtualBox option is the most suitable for end-users. The Docker is also a straightforward option for people who are familiar with containers. The installation scripts is tedious and error prone method as it requires that the init scripts are maintained for many different Linux distributions and assumes knowledge of several Grid services for troubleshooting installation issues.

Once the environment is configured, a user can start with the VM management. The basic VM management tool is OCCI. OCCI is a protocol and API for IaaS remote management. OCCI provides a uniform interface for integration and interoperability across sites that run different Cloud management software. OCCI supports both cloud stacks available at SURFsara: OpenNebula and OpenStack.

In order to spawn a VM from the command-line, a user has to follow all these steps: setup a machine with the Grid-based FedCloud environment, install his Grid certificate, discover the appliance IDs from the AppDb or by querying the BDII information system with an LDAP client, find the endpoint URLs for the sites that support the appliance, setup a contextualization script to access the deployed VMs, launch the VM and attach a network interface and a storage datablock with an OCCI client. Starting a VM returns back a unique VM ID that is used to manage it.

A command-line OCCI client is rOCCI (ruby based). The rOCCI command structure is:

occi --endpoint A --auth B --action C –resource D

where,

A: site endpoint URL

B: x509 proxy

C: actions like create/stop/describe VM, attach/detach a network/datablock, list VMs, etc.

D: compute, storage

mixins: define the type of templates (OS or Resource template).

Here is an example of an rOCCI command to launch a VM on a Spanish FedCloud site:

occi --endpoint http://server4-epsh.unizar.es:8787 --action create --resource compute --mixin os\_tpl#2cc50cbc-8694-42f9-a3ed-0021fd0004b1 --mixin resource\_tpl#m1-small --context user\_data="file://$PWD/tmpfedcloud.login" --attribute occi.core.title="Sara VM" --auth x509 --user-cred /tmp/x509up\_u500 --voms

returned URL

> http://server4-epsh.unizar.es:8787/compute/2048be92-dce6-4f62-8278-1fd7babcd1ba

Besides the command line client, there are several APIs ready to be used with the EGI Federated Cloud: OCCI API with jOCCI (Java SDK) or rOCCI (ruby SDK) & OpenStack API for OpenStack sites of the EGI Federated Cloud.

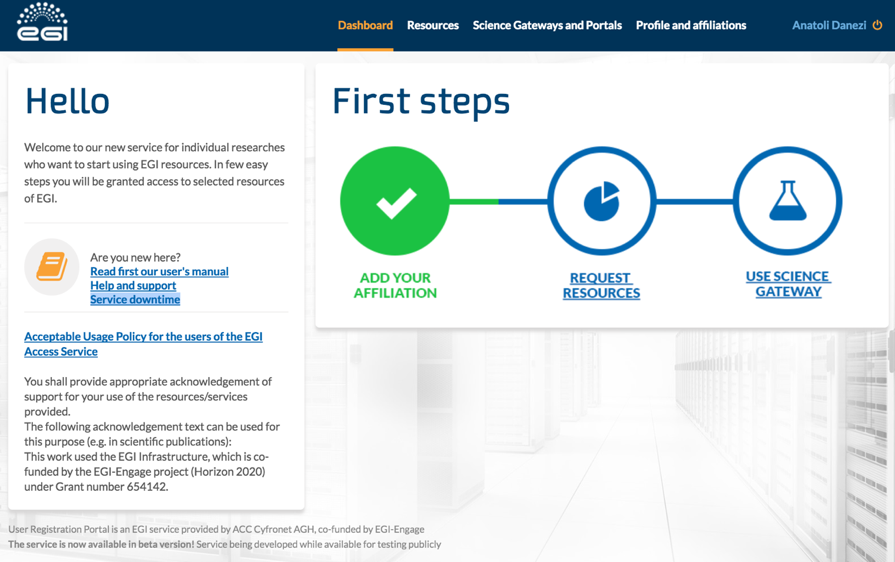
It is worth to mention here that lately there are discussions in EGI about stop supporting the OCCI interface as the unifying API of Federated Cloud and promote the direct use of native interfaces under a Cloud Management Platform (e.g. Scalr, SlipStream, Terraform, Cloudify, IM, OCCOPUS, INDIGO-DataCloud). Some reasons to stop supporting OCCI are incompatibilities with cloud stacks, lack of adoption, little support for the CDMI object store API and the fact that commercial providers see no added value in OCCI.

### Applications on Demand

Applications on Demand ([AoD](https://access.egi.eu/start))(*Beta*) is a service announced recently that is very promising for the 'long tail of science'. It is designed to offer easy access to a portfolio of scientific applications. The following components are already integrated into the service as taken from the EGI website:

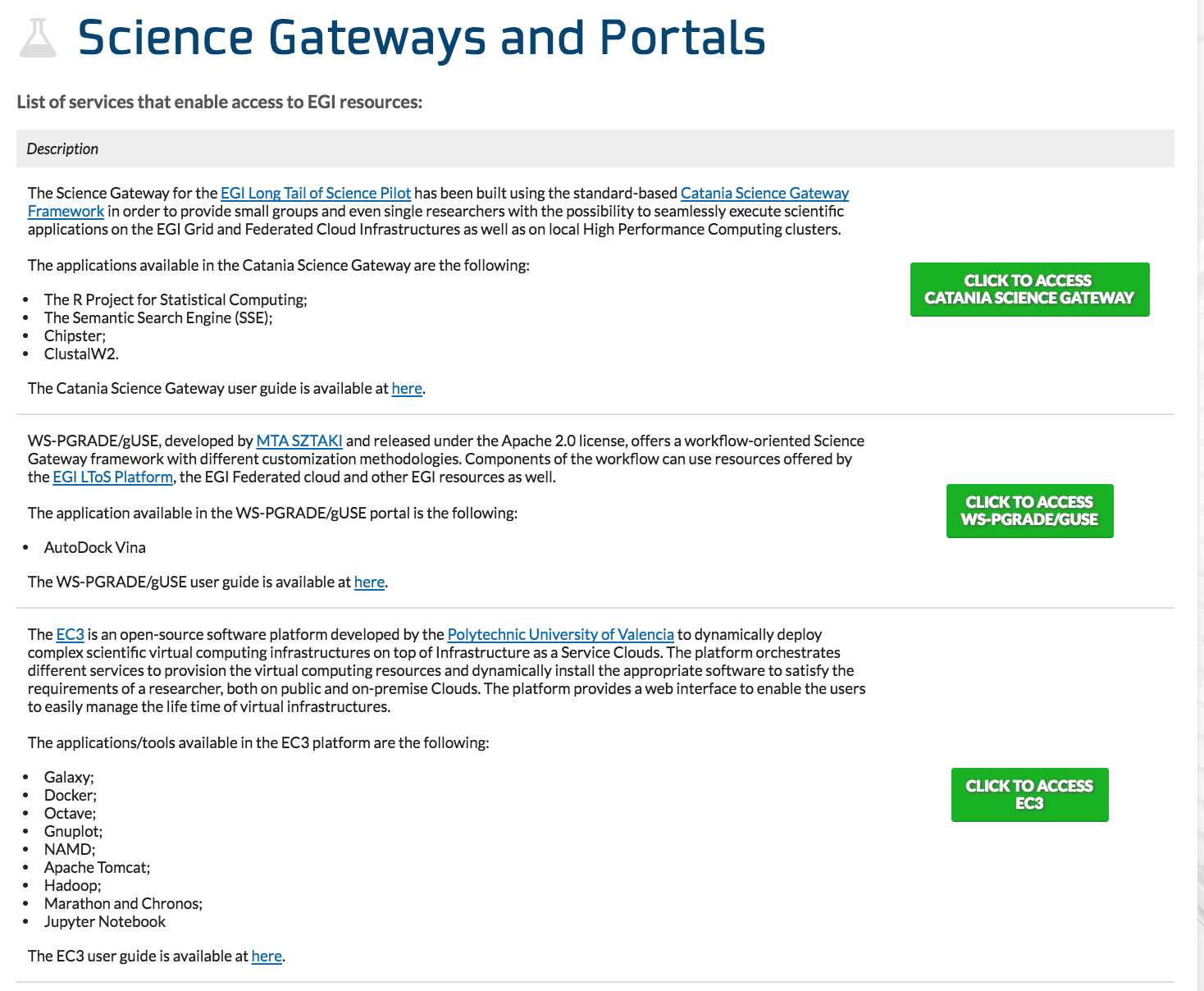


AoD’s compute and storage services are currently provided by CYFRONET, INFN, BELSPO, MTA SZTAKI, CESGA, UPV, and BIFI. The service webpage is well presented with clear instructions on what it offers, for whom, and how to register to the service. Access to the service is based on EGICheckIn and does not require a Grid certificate per se. We used SSO authentication for our tests. We also tried X509 authentication which failed directing us to a landing page written in polish. However, we were able to request for resources easily by navigating in the service dashboard:



The response and approval from the EGI support team was very fast. For each resource request a unique ID is generated with a reference to the SLA.

Once the resources are granted a user gets has access to Science Gateways and Virtual Research Environments: EC3, Catania gateway, WS-PGRADE gateway. Each has a readily available set of applications as shown in the picture below.

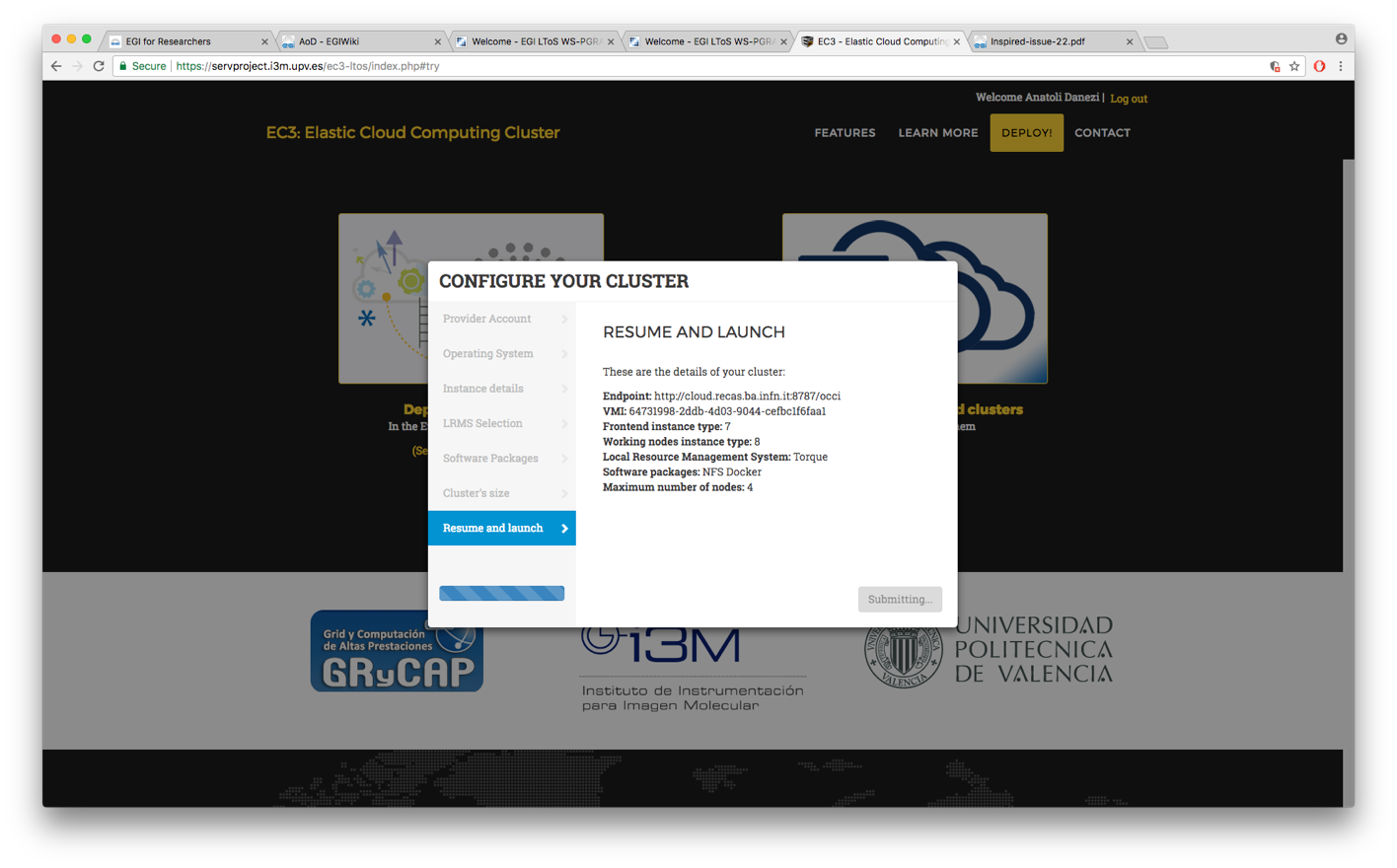


We managed to get a working environment with the [EC3](http://servproject.i3m.upv.es/ec3/) service. EC3 is a platform which allows to create elastic virtual clusters on top of Infrastructure as a Service (IaaS) providers. We highlight here a few of the applications:

* Jupyter Notebook: a VM pre-installed with Jupyter Notebook on EC3.
* Docker: a VM with Ubuntu or CentOS that installs Docker at first boot.
* Chipster: a VM with Ubuntu or CentOS that installs Chipster at first boot.
* Galaxy: a VM with Ubuntu or CentOS that installs Galaxy at first boot.

The cluster, which is defined with a 'wizard' interface for the user, is composed by a front node, where a batch job scheduler is running, and a number of compute nodes. The list of application libraries/tools installed in the front-node can be exported via NFS in all the compute nodes. The installation and configuration of the cluster is performed with Ansible.

The following picture shows the EC3 interface when deploying a Docker cluster:



We deployed several clusters with the EC3 'wizard', such as Jupyter Notebooks, Docker and Chipster. After a successful deployment we received an IP, a username and an ssh key to login to the front node and configure it. There are also ready-to-use recipes to start Docker Swarm or Kubernetes clusters on EGI Federated Cloud via the AoD.

We found that the elastic cluster deployment works out-of-the-box, although the user needs to select on which FedCloud site he wants to deploy a certain application, while we would expect that the infrastructure hosting the applications would be coherently unified and transparent to the AoD user.

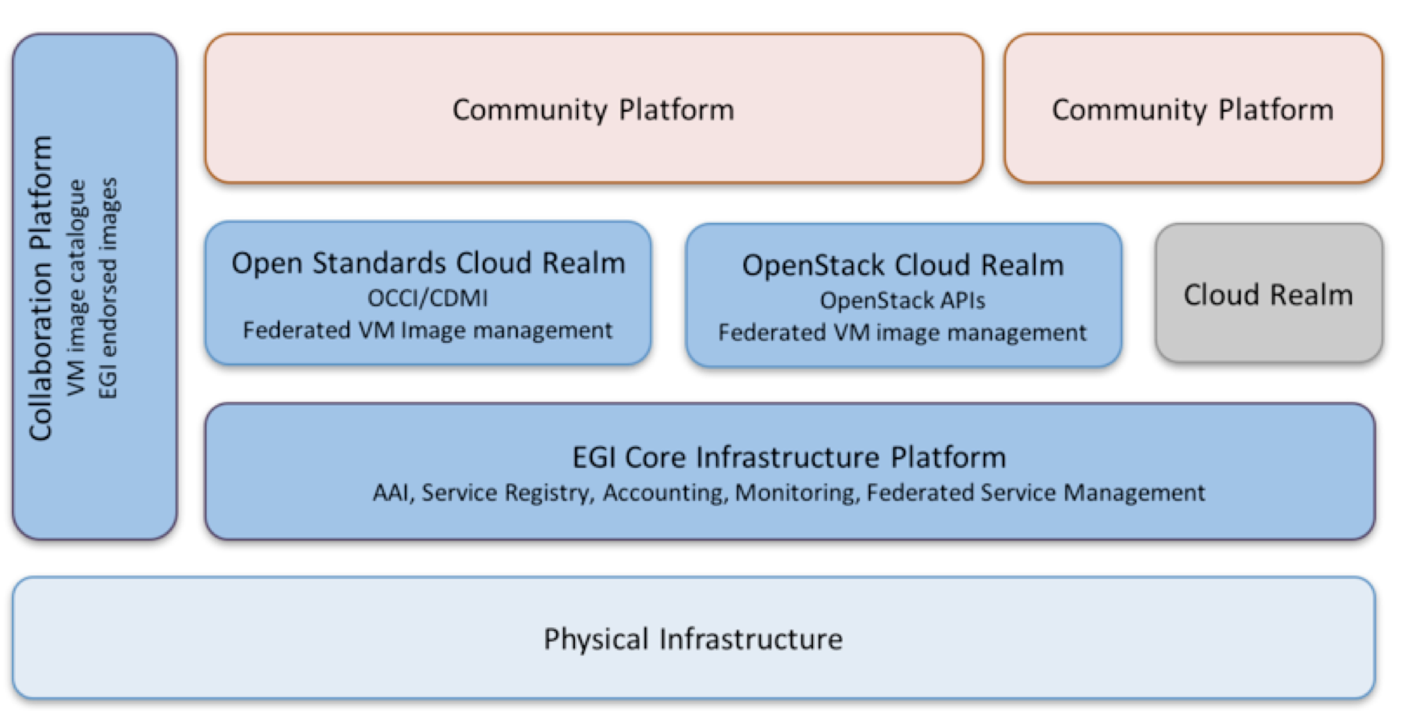
## Resource providers

The EGI Federated Cloud resource providers are institutions and companies contributing to the federation by providing access to their cloud infrastructure. Currently the EGI cloud federation includes 24 cloud sites from all across Europe. Most of the sites (17) are based on OpenStack.

Each resource provider integrates with the federation by installing the following EGI core components:

* Information Discovery: for configuration management of federated cloud services.
* Accounting: for collecting, and displaying usage information.
* VM Image Management: for replicating VM images as needed by the user communities in a secure way.
* Federated AAI: for authentication and authorisation across the whole cloud federation.
* OCCI interface
* Monitoring: for performing service availability monitoring.

In the previous evaluation document, we described in detail the tools that have to be installed to support the operations services.



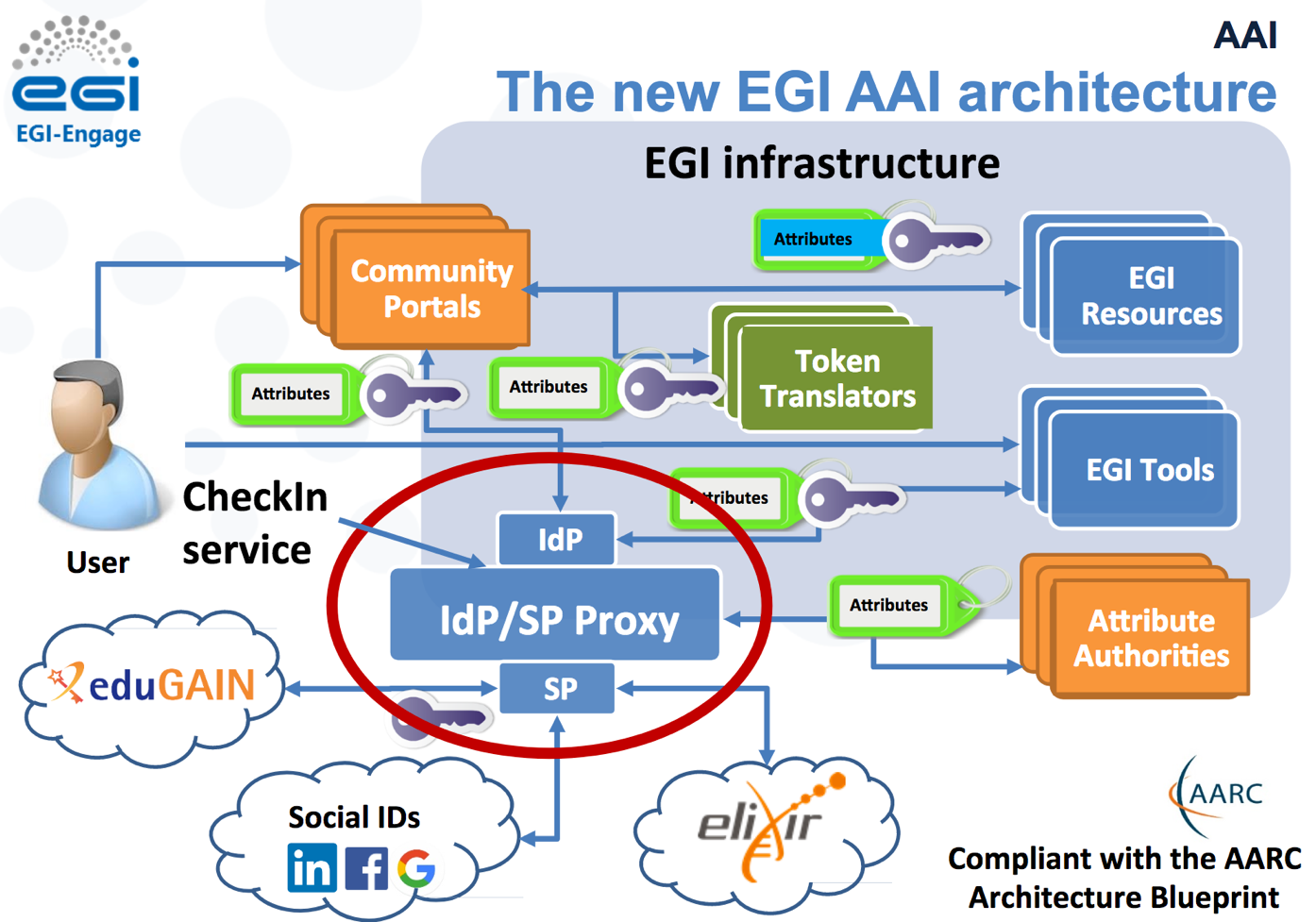
For OpenStack, information discovery, accounting and VM Image management components can be run on a single VM that encapsulates them for convenience. Federated AAI requires installation of a plugin in Keystone, and OCCI interface is installed alongside the nova-api.

In OpenStack, the components of the private cloud that interact with the EGI Federated Cloud components are: Keystone, Nova, Ceilometer and Glance.

According to the EGI operations site, it takes about 3-5 working days to install the needed services on a new site. Every operating site needs to go through 3 days of operation to qualify for the certified state. We believe that the integration barrier is much lower for OpenStack providers than for OpenNebula or other cloud stacks.

## Authentication and authorisation

Authentication on the EGI Federated Cloud is handled by the EGICheckIn service which uses federated authentication mechanisms. It connects federated Identity Providers (IdPs) residing ‘outside’ of the EGI ecosystem, see the AAI architecture in the picture below:



We see that EGI has several ways you authenticate you. Available IdPs are: EGI SSO, ELIXIR AAI, eduGAIN, X509 (IGTF), Social Networks (Facebook, Google, LinkedIn), and more (ORCID, VHO). For instance, the SURF members can access many EGI services via eduGAIN. However, all these different authentications for the same person do not seem to integrate well. We experienced several authentication issues during the testing of the different services.

The authorisation to the resources is based on Virtual Organisations (VOs). In practice a VO membership determines to which resources (compute and storage) a user has access to. Each VO includes a subset of the EGI Federated Cloud sites, and makes those available for the given community through generic and/or community-specific policies and protocols. There are generic VOs, for example the fedcloud.egi.eu VO, open for any user who wants to experiment with the EGI Federated Cloud and domain-specific VOs like vo.elixir-europe.org for the Elixir community members who want to run applications developed by the Elixir service providers.

## Data management

On EGI Federated Cloud, the OS images are non-persistent by default, thus a Block or Object store is required for the users to save their data. However, we found that the data blocks stored on a site cannot become easily available to another site of the federation although it is very easy to create and attach them to a VM via the VMops portal.

An alternative to the Block storage is the use of Object Storage. EGI currently offers two APIs for object store: SNIA Cloud Data Management Interface (CDMI) which defines a RESTful open standard for operations on storage objects, and SWIFT which is the OpenStack Object Store project, also providing a RESTful API for managing and accessing the objects. CDMI clients are still very poor and SWIFT sites are only a few in the federation.

As an interface to the data storage, EGI promotes the EGI DataHub service that promises easy and scalable access to data from federated cloud sites. In our experience the Onedata technology behind the service is not workable for data-intensive computing. Onedata aims to integrate a wide range of existing storage services, regardless of their underlying technology (e.g. Lustre, Amazon S3, Ceph, NFS, or dCache) which is hard to apply in practice for all the different storage endpoints.

There are some other initiatives to provide solutions for the lack of clear data management solutions, such as an EGI-EUDAT Integration pilot for processing long-term storage in EUDAT (B2stage/B2safe) and other domain specific solutions offered e.g. by Elixir. There is ongoing effort in Elixir community to provide a dataset replication engine accessible from the EGI Federated Cloud for the reference datasets.

# Evaluation of support and usability

## Documentation

The user documentation is in a better state than during our first evaluation in 2014. However the information is still scattered and not easily findable especially for non-expert users. The same applies to the documentation for resource providers.

Questions still open to us are:

* Are the workflows and roles of the resource providers and EGI defined and documented?
* Is there clear documentation for the resource provider on how to support new VOs?

## Enforcement

There are no measures to check running VMs against the validity of the user (account terminated) or the user's proxy certificate (expiration). When a user forgets a VM, it will/might continue to run indefinitely. Compare this to CSC's (Finish NGI) approach: each VM needs an expiry date after which it can be destroyed.

Questions still open to us are:

* How much storage do we need to reserve for guest user data as resource providers?
* When can the guest data be deleted?

### Accounting and billing

We have no (or not enough) experience with the way accounting and billing is done on the EGI Federated Cloud and cannot judge [if the existing portal](https://accounting.egi.eu/) is useful for our own (future) procedures. This applies both to other users using our facilities and to our users using other facilities.

Questions still open to us are:

* Do we want to bill other users directly or send a total to each provider?
* Do we want to receive one big bill for all our user's time on another provider?
* Can we trace individual users' usage on other providers?

## SLA, SW ownership

Federated resources need managing and clarity in ownership and responsibility of software, hardware and services.

Questions still open to us are:

* What SLA do we have to adhere to when joining the EGI Federated Cloud?
* Who is responsible that an application works as expected on our cloud site?
* How are the updates synchronised with other services?
* Will it be clear who handles some category of helpdesk tickets?

# Conclusions

## Overall

During the first evaluation of the EGI Federated Cloud in 2014 we found that deploying and accessing even a stock VM took much time and effort because of the complex environment and inconsistencies between the different site endpoints. The federated sites were not unified coherently in a single system as the endpoints had to be known in advance, which in turn made the cross-site VM deployment not transparent to user.

Moreover, back then, the lack of documentation and non user-friendly environment together with the Grid certificate dependencies put a lot of burden to the user. Also, there were no policies in place to help the smooth entry of a new provider to the federation. For example, information about accounting, monitoring and service level description was mostly missing.

However, today the new EGI Federated Cloud features add remarkable improvement as it seems that the communities and resource providers' feedback has been incorporated. As example, the new services of VMops and Application on Demand (AoD) implement much clearer interfaces between the different parties: the individual scientists and community members who make use of the resources, the application developers who take care of the VMs configuration and the service providers who arrange the distribution of the VMs and the applied policies. Previously, all of these tasks had to be handled by the end-users making EGI Federated Cloud impossible to use.

We would like to stress that during the first evaluation as well as the latter one, we always received very fast reaction from the EGI support team to all of our service requests.

Still, there are several bugs to fix, especially in the AAI methods. More importantly, the EGI Federated Cloud long-term vision is not entirely clear. It is not clear whether the EGI Federated Cloud services would act as an intermediary distributing VMs among sites (Broker) or keep the concept of integrating different Cloud sites in a way transparent to the user (Federation).

From the user perspective, the VM management has been simplified a lot with the VMops dashboard that effectively hides the backend complexity. However, we detected a lack of clear model for the data management. Data stored on a site cannot become easily available to another site of the federation and the existing APIs for object storage are still very poor. EGI promotes the DataHub service that promises easy and scalable access to data from the federated cloud sites, but in our experience the Onedata technology behind the service is not workable for data-intensive computing.

From the resource provider perspective, the sites need to adhere to a set of common policies, procedures and processes for federated service management, although there is no guarantee of QoS at the moment. However, EGI offers a baseline for operational policies, but leaves the freedom to build an SLA between the site provider and the supported community (VO) in order that the provider's service level is not affected.

In overall, we have a positive impression for the developments in the EGI community because the design and implementation of the new services (VMops and AoD), as well as the upcoming (Marketplace) shows that the EGI community refined the initial concept and evolved its architecture according to emerging user demands. Not all of the services are in a state to support production runs for large-scale intensive computing, but the scope and design of the new services is towards the correct direction.

## Considerations for adoption

The decision for SURFsara joining the EGI Federated Cloud as a resource provider requires planning, thorough estimation of the effort required for manpower and resources, balanced to the benefits for the Dutch research communities and our company's political placement in the international research support. Thus, the decision can be only taken in consultation with our management. Hereby, we present the technical implications and potential user benefits to help this decision.

### Technical implications

Currently SURFsara employs two types of Cloud systems: OpenNebula operated by the HPC Cloud and OpenStack operated by the DDP team. If we decide to become part of the federation, we would recommend joining with OpenStack and not OpenNebula for technical reasons, as well as human resource efficiency.

On the technical side, we found that the integration of OpenStack sites is more smooth than OpenNebula. OpenNebula implementation has several limitations for embracing properly some EGI services, such as the EGI AAI mapping. In opposite, OpenStack backend is more compatible and the available configuration documentation is more extensive.

On the human resource efficiency side, we believe that the system administrators who support our current DDP OpenStack systems have extensive knowledge on several Grid services which are also used in the EGI Federated Cloud implementation. Moreover, the same people are involved in the initiative between SURFsara and and RUG/CIT for federative services based on OpenStack.

The EGI's OpenStack integration will need to be tested to ensure compatibility with our OpenStack configuration and avoid unexpected issues with our other services. This also includes testing OpenStack release dependencies. The production-level integration is not trivial, and this needs to be scheduled into the operators' development work. Operating an OpenStack cloud is resource intense. Adding extra integrations to a production OpenStack cloud would increase the complexity of maintaining the cloud.

Moreover, we should not neglect the effort that will be required for support. A resource centre part of the EGI Federation, and supporting international communities, needs to provide support through the EGI channels. This means following up tickets submitted through helpdesk.egi.eu. This includes requests from user communities and tickets triggered by failures detected by the monitoring infrastructure.

The estimated integration time would be 2 weeks minimum for two system administrators with expertise in EGI and Grid services. We estimate minimum of 0.5 fte to maintain sufficient technical knowledge, keep things running and up to date, and providing basic support. Providing tailored support and consultancy, and keeping things connected on the management level would easily become more than 1 fte.

Lastly, we should stress here that the EGI Federated Cloud implementation allows anonymous users gaining superuser rights on the host hardware who may not behave. Thus, if we join the federation we need to ensure that the full project lifecycle is clear in the SLA, e.g. what happens with data if researchers leave institute or the VO or how is the traceability for security incidents arranged. On the SURFsara Clouds, we use project-based access control where all resources are shared within a project. The same can be potentially done for a certain VO group. It is possible for us to support certain VOs for approved projects and set quota per project on our side. For instance, we can support appliances and virtual organisations required by the ELIXIR Compute platform. Agreements can be tailored as 1-1 negotiation between the site provider and use case.

### Value for user community

One of the main goals of evaluating the EGI Fed Cloud is to find whether their services can add any value to the needs of the SURFsara users as we believe that the decision for adoption should be taken with/for our communities.

Implementing EGI FedCloud at SURFsara should make enough sense for one or more user communities that we support, for example Elixir. Although no production runs are expected for Elixir in the short term, they are currently testing workflows, AAI methods and data management models integrated with the EGI Federated Cloud.

Another added value would be joining the federation to support multinational projects when local capacity is not sufficient (e.g. scale out on peak demand). However, it is doubtable whether the current state of the EGI Federated Cloud can support such big projects. The total resource capacity is quite steady over the past years. The total capacity is 6,600 cores and few hundred TBs storage provided by all sites together. We find that applications needing considerable amount of resources in term of computation and/or memory and/or intensive I/O cannot be accommodated with the current EGI Federated Cloud capacity.

Further, we see an added value for users who want to run applications that are currently not in our service portfolio, such as: web hosting, Docker container compute, Chipster or Galaxy. JupyterHub is offered both by SURFsara and EGI which could also be investigated further for a potential collaboration. In general, we find that the new services (VMops, AoD, Marketplace and EGICheckIn) and the policies and processes built by the EGI community for the Federated Cloud can very useful for many community service providers who develop their own services for a wide range of users.

# References

* [SURFsara website](https://www.surf.nl/en/about-surf/subsidiaries/surfsara/)
* [EGI website](https://www.egi.eu/services/)
* [FedCloud user wiki](https://wiki.egi.eu/wiki/Federated_Cloud_user_support)
* [AppDB Cloud Marketplace](https://appdb.egi.eu/browse/cloud)
* [VMops dashboard](https://dashboard.appdb.egi.eu/vmops)
* [VMops blog article](https://www.egi.eu/about/newsletters/a-new-interface-for-the-egi-federated-cloud/)
* [AoD user wiki](https://wiki.egi.eu/wiki/Applications_on_Demand_Service_-_Information_pages)
* [EGI CheckIN](https://www.egi.eu/about/newsletters/egi-services-for-open-science/)
* [FedCloud resource providers](https://wiki.egi.eu/wiki/Federated_Cloud_resource_providers_support)
* [Slides Tiziana Ferrari, EGI conference 2017](https://documents.egi.eu/public/RetrieveFile?docid=2184&version=1&filename=EGI%20keynote%202017%20short.pdf)