Infrastructure Manager

Ondřej Melichar, Josef Handl

September 2020

1 Introduction

This document serves as a brief overview of the Infrastructure Manager tool and its basic capabilities and functionalities. The following information are presented as understood from the use of the Infrastructure Manager at the Institute of Computer Science, Masaryk University.

The Infrastructure Manager was developed by the Grid and High Performance Computing Group (GRyCAP) at the Instituto de Instrumentación para Imagen Molecular (I3M) from the Universitat Politècnica de València (UPV). It serves as an interface for the management of various virtual machines or complex infrastructures. It supports multiple APIs from various cloud platforms and overally eases the process of deployment, configuration, software installation, monitoring and update of the machines in question. The tool also features a XML-RPC API and a REST API [1].

2 Access

2.1 Virtual Organization

Infrastructure Manager uses EGI authentication mechanism called *EGI Check-in*. This is similar to *ELIXIR*. In order to be able to acces the Infrastructure Manager and more importantly, the EOSC Synergy resources, one needs to apply for the membership in the *eosc-synergy.eu* virtual organization. The VO and all its information can be seen here and the enrollment process starts here. Once approved, it is possible to start using the Infrastructure Manager tool. In case of any issues with the approval, contact Markus Hardt at hardt@kit.edu.

2.2 Sites

There are two sites from which the service can be accessed from. Each of these sites offer a slighty different set of tools and complexity. The first one is the more modern *IM-Dashboard* and offers simpler user interface but less possible configuration settings. The second one is *IM-web*, which has a slightly different interface and offers more complex configurations. As a main configuration file,

the Infrastracture Manager uses the RADL and TOSCA formats which are used to describe the desired infrastructure. More to TOSCA can be found here, where as information about RADL can be found here. The beforementioned configuration file is called a *recipe*.

3 IM-Dashboard

As we at ICS use IM-Web only, this chapter will only outline few basic options and mechanisms. The site itself is available at this link. Apart from the graphical user interface, the main difference from IM-web is the limited number of pre-defined recipes, which are offered to the user, as shown in 1.

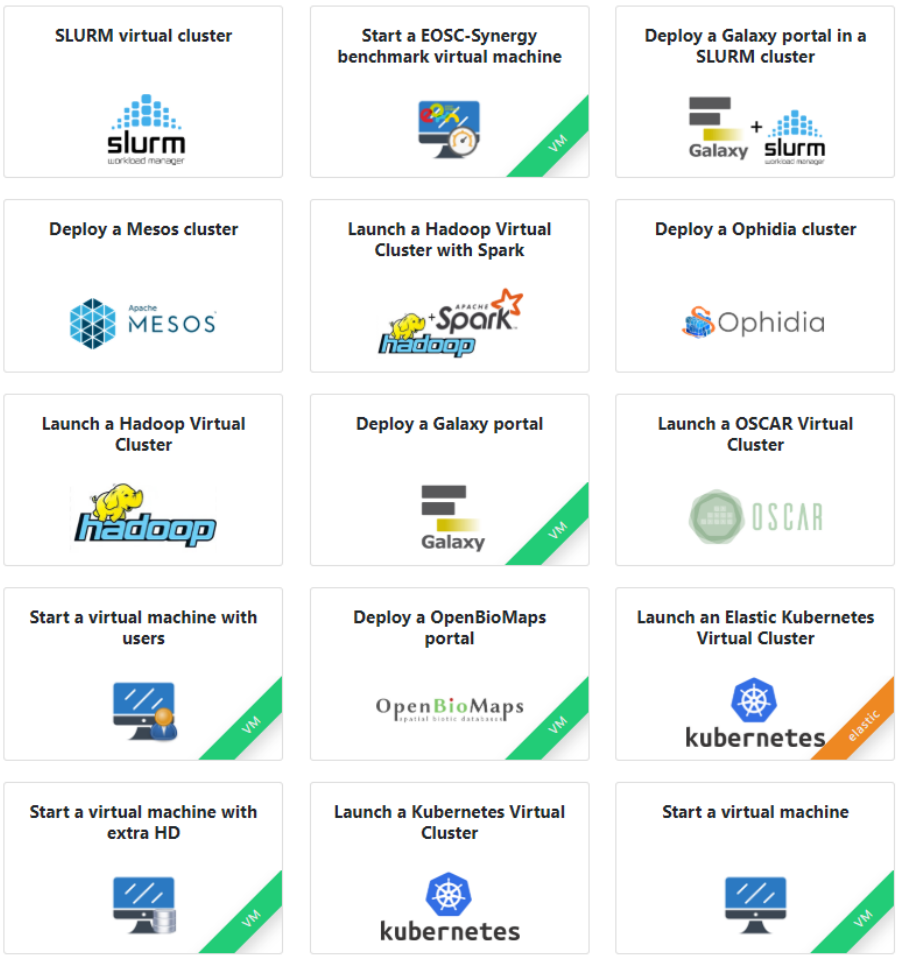

Figure 1: List of predefined recipes

The possibility to customize these recipes is limited and there is no way to directly add a custom recipe, as shown in figure 2.

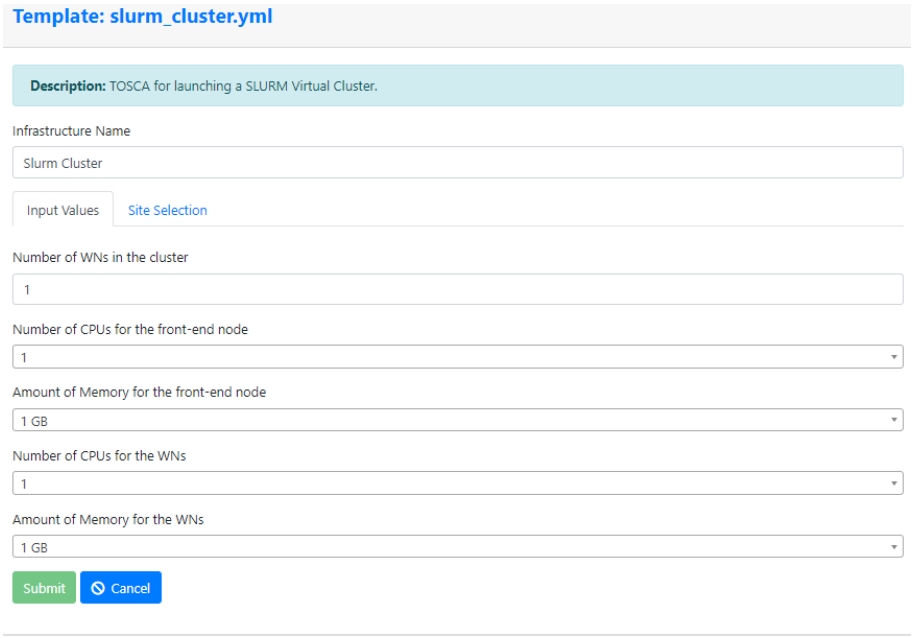

Figure 2: Limited configuration options

The user also needs to specify a site in which the infrastructure should be deployed to. Upon selecting the *eosc-synergy.eu* VO, a list of possible resource providers is generated and the user can choose one of them. Then, the disk image needs to be chosen. This is also related to the site which the user chose, as shown in figure 3

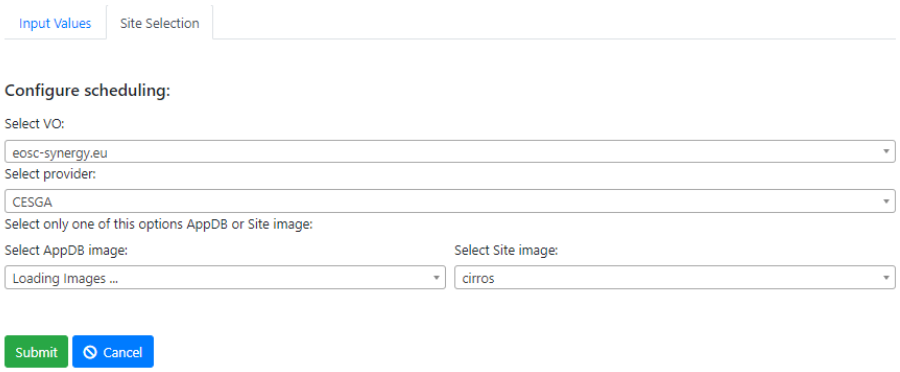

Figure 3: Site selection

After launching, user can see the state of the infrastructure in the *Infrastructures* section. There, the user can delete the infrastructure, perform reconfiguration or see the logs. The information about the assigned IP address or adresses is also present along with the access credentials.

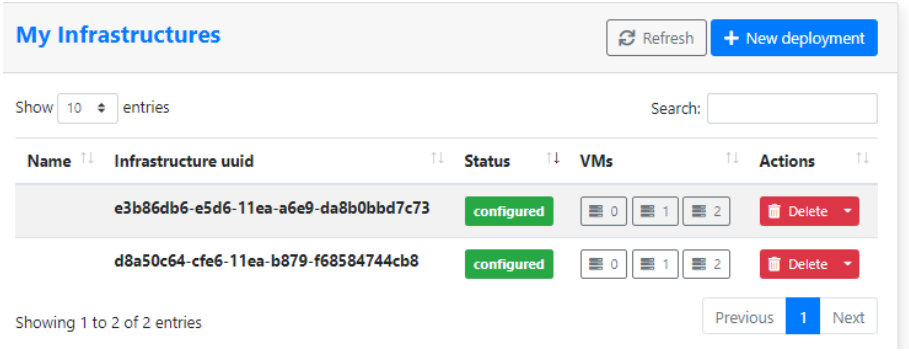

Figure 4: Infrastructures overview

The content of the IM-Dashboard recipes themselves is not visible to the user. IM-Dashboard supports only the TOSCA formatted recipes. The actual content of these TOSCA templates can be found in the project's GitHub repository.

4 IM-Web

The IM-Web application is available at this link. The authentication is again possible via EGI Check-in. There are three sections which the user can click on.

- Infrastructures
- Credentials
- Topologies

The *Infrastructures* section offers the same options as IM-Dashboard - management, reconfiguration, logs etc. The *Credentials* section stores the user's credentials for various resource providers (CESGA, CESNET..). For each site the user wants to deploy in, there needs to be a record. There records can then be placed in a desirable order. The highest (first from the top) record will have the biggest priority and the recipes will be deployed in that particular site. In the table 1, there's an example of a filled-in credentials record for the CESGA site, which uses OpenStack technology.

ID	CESGA
Host	https://fedcloud-osservices.egi.cesga.es:5000
Tenant	openid
Auth Version	3.x_oidc_access_token
Base URL	-
Username	egi.eu
Password	-
Domain	-
Region	-

Table 1: Example of a filled credentials record

After having a valid credentials record, it is possible to deploy a machine in the particular site. First, a recipe needs to be created though. In the IM-web application, there is also a number of default templates available, but unlike IM-dashboard, it is possible to create a custom recipe in the RADL or TOSCA format. There recipes are stored in the *Topologies* section. An example of a simple ubuntu 18.04 VM with 2 CPU cores, 4 GB of RAM is located below. The recipe is in RADL format and contains two additional tasks in the *configure* section. The configure sections serves as a place to add custom Ansible tasks, which are executed during the deployment. In this case, Ansible will gather information about all nodes and in the second taks, it will install two packages. The *deploy* keyword is followed by the node definition name and a number of VMs of this type that should be deployed.

```
network pub (
   outbound = 'yes'
)

system node (
   cpu.count >= 2 and
   memory.size >= 4G and
   net_interface.0.connection = 'pub' and
   disk.0.image.url = 'appdb://CESGA/egi.ubuntu.18.04?eosc-synergy.eu'
)

configure public_node (
@begin
---
- hosts: node
  tasks:
    - name: Gather facts for all nodes
      debug:
        msg: "All facts gathered"
```

```
    - name: Install packages
      package:
        name: ['python3-pip', 'libpq-dev']
@end
)

deploy node 1
```

Once the recipe is saved and the credentials records are in proper order, the VM can be launched. If successful, it will be in the *configured* state upon finishing. Otherwise it will remain *unconfigured* and it is necessary to go over the logs to find the issue. The detail of a configured VM can be seen in figure 5. The full information about the VM status is present there and also a link to download the access key.

Figure 5: VM detail

5 Conclusion

The beforementioned facts are based on a subjective experience. There's additional functionality that isn't covered by this document. Additional useful information can be found in the project's documentation page and in the related literature [2].

References

[1] M. Caballer. Im documentation 1.0. Available at https://imdocs.readthedocs.io/en/latest/index.html.

[2] M. Caballer, I. Blanquer, G. Moltó, and C. de Alfonso. Dynamic management of virtual infrastructures. *Journal of Grid Computing*, 13(1):53–70, 2015.