INFN Cloud Dashboard

Marica Antonacci (INFN BA)

Uso e sviluppo di applicazioni e servizi su INFN Cloud (CLueApp)

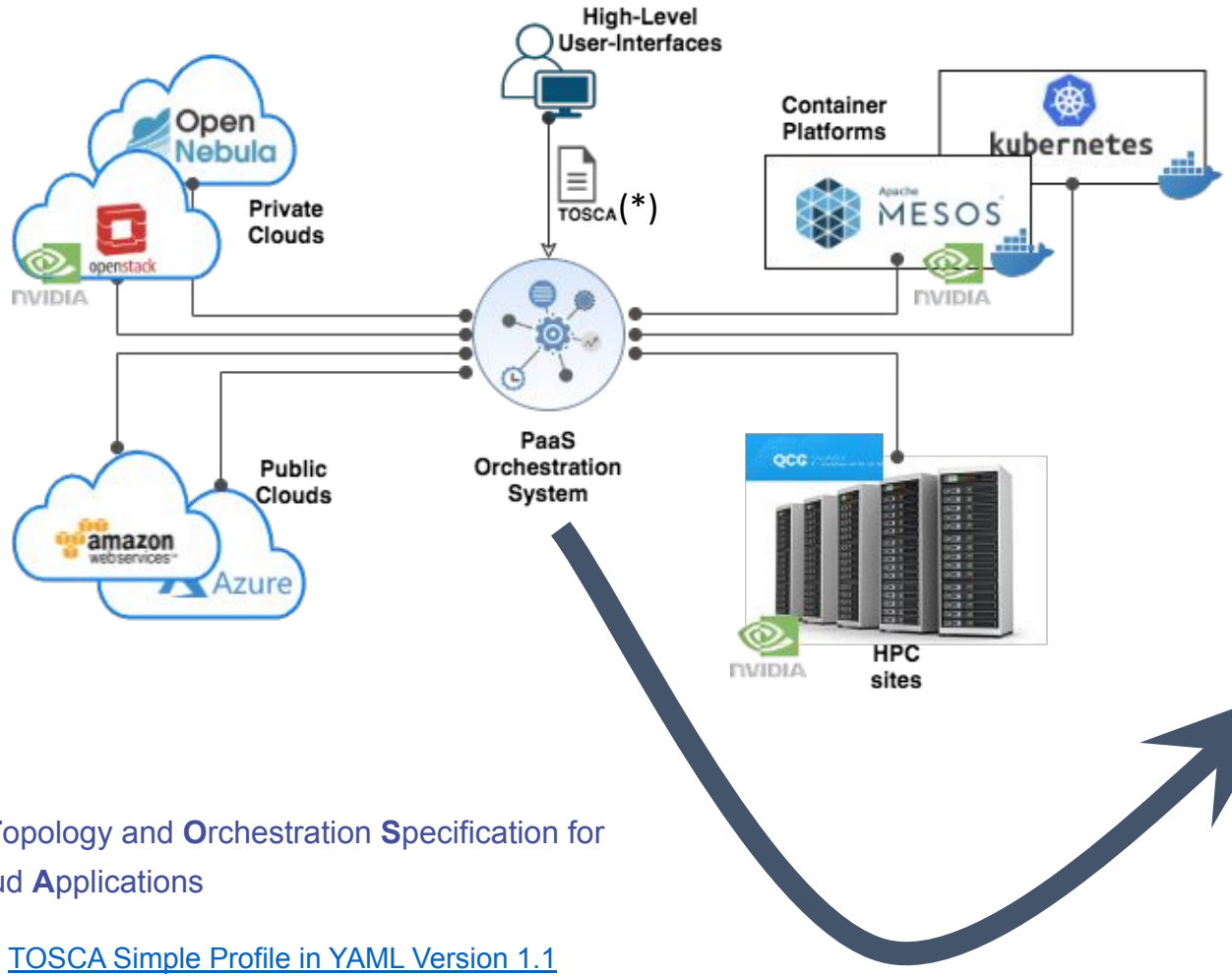
13-16 September 2022

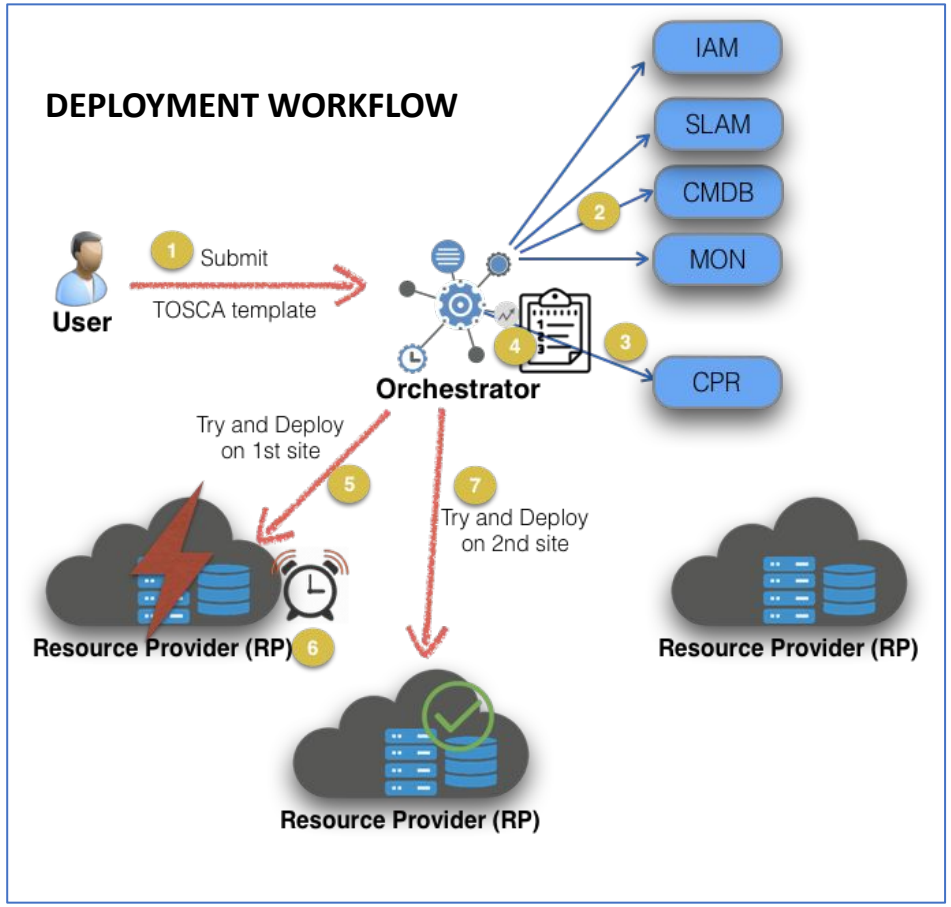
Marica Antonacci (marica.antonacci@ba.infn.it)

The INFN Cloud

- **INFN Cloud aims to offer a full set of high-level cloud services to INFN user communities**
  - the service catalogue is not static: new applications are included through a defined “on-boarding” process for new use-cases
- **Architecturally INFN Cloud is a federation of existing infrastructures**
  - the *INFN Cloud backbone*, consists of two tightly coupled federated sites: BARI and CNAF
  - a scalable set of satellite sites, geographically distributed across Italy, and loosely coupled.
- **Key enabling factors for the federation**
  - leverage the same authentication/authorization layer based on **INDIGO-IAM**
  - agree on a consistent set of policies and participation rules (user management, SLA, security, etc.)
  - transparent and dynamic orchestration of the resources across all the federated infrastructures through the **INDIGO PaaS Orchestrator**

PaaS Orchestration System (from 10Km)





(*) **T**opology and **O**rchestration **S**pecification for **C**loud **A**pplications

Ref: [TOSCA Simple Profile in YAML Version 1.1]

Marica Antonacci (marica.antonacci@ba.infn.it)

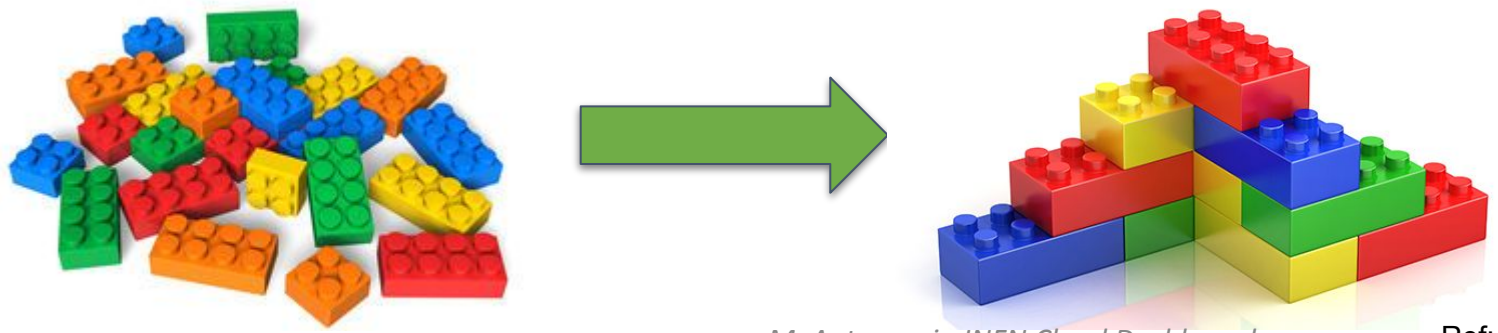
The INFN Cloud services

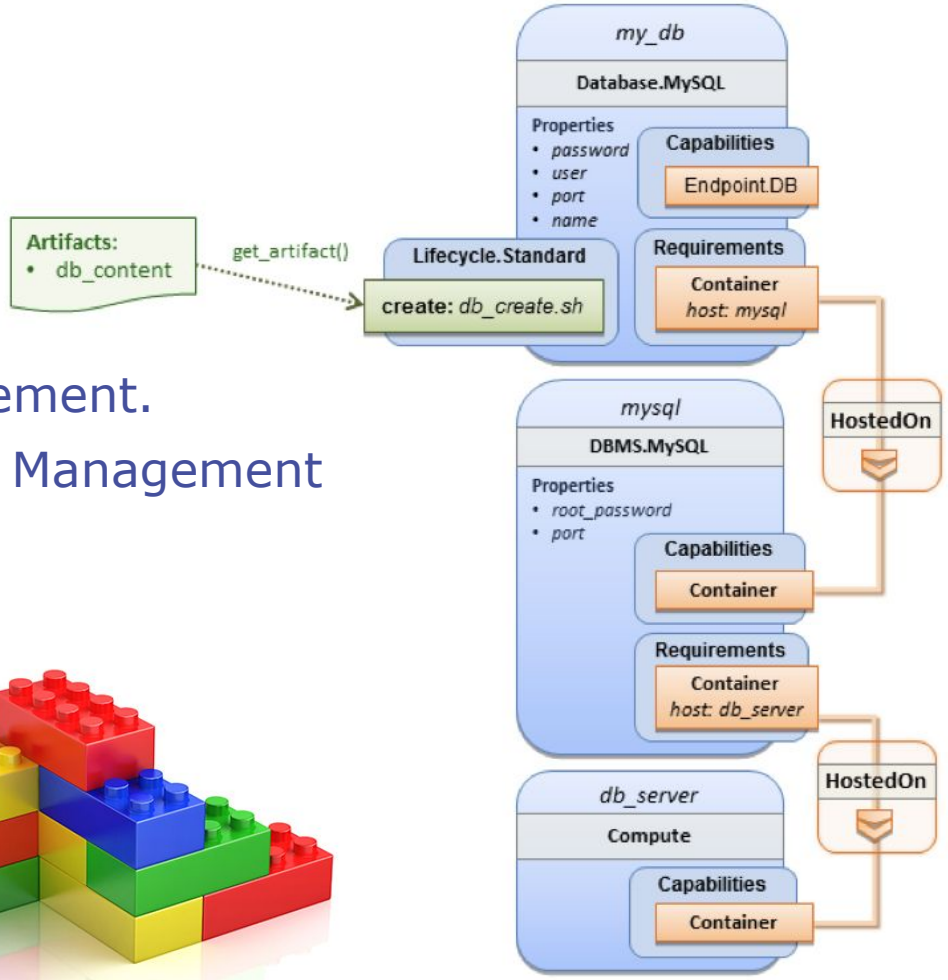
- The INFN Cloud services are based on **modular components and span the IaaS, PaaS and SaaS models** for both computing and data.
- All services are described by **TOSCA templates** (which can refer internally to other components such as Ansible playbooks, HELM charts, etc.).
- The services can be **deployed** via the INFN Cloud Dashboard or via a command line interface:
  - **Automatically** by the INFN Cloud Orchestrator on one of the federated Cloud infrastructures, depending on resource availability and policies.
  - **Manually** by a user on a specific federated Cloud infrastructure.

TOSCA

Topology and **O**rchestration **S**pecification for **C**loud **A**pplications

- Goals:
  - Automated Application Deployment and Management.
  - Portability of Application Descriptions and Their Management
  - Interoperability and Reusability of Components





Ref: TOSCA Simple Profile in YAML Version 1.1

Marica Antonacci (marica.antonacci@ba.infn.it)

Template example

```
tosca_definitions_version: tosca_simple_yaml_1_0_0

description: Template for deploying a single server with predefined properties.

topology_template:
  inputs:
    cpus:
      type: integer
      description: Number of CPUs for the server.
      constraints:
        - valid_values: [ 1, 2, 4, 8 ]

  node_templates:
    my_server:
      type: tosca.nodes.Compute
      capabilities:
        # Host container properties
        host:
          properties:
            # Compute properties
            num_cpus: { get_input: cpus }
            mem_size: 4 MB
            disk_size: 10 GB

  outputs:
    server_ip:
      description: The private IP address of the provisioned server.
      value: { get_attribute: [ my_server, private_address ] }
```

```
tosca_definitions_version: tosca_simple_yaml_1_0_0
description: Template for deploying a single server with MySQL software on top.

topology_template:
  inputs:
    # omitted here for brevity

  node_templates:
    mysql:
      type: tosca.nodes.DBMS.MySQL
      properties:
        root_password: { get_input: my_mysql_rootpw }
        port: { get_input: my_mysql_port }
      requirements:
        - host: db_server

    db_server:
      type: tosca.nodes.Compute
      capabilities:
        # omitted here for brevity
```

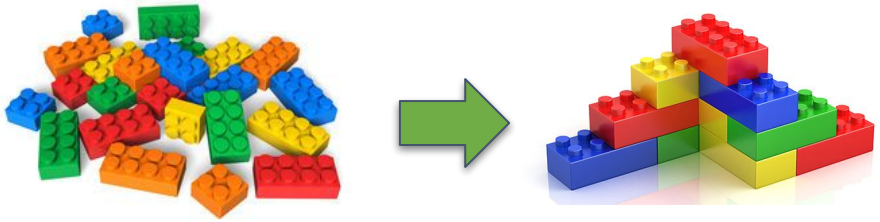
The service catalogue

The catalogue is a graphical representation of the TOSCA templates repository that we have been developing extending the INDIGO-DC custom types

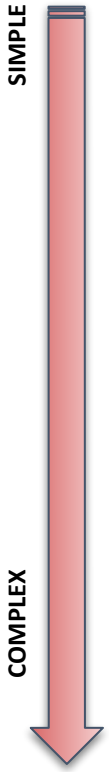
- Each card in the catalogue is associated to one or more templates
- We are following a **lego-like** approach, building on top of reusable components and exploiting the TOSCA service composition pattern

Main objectives:

#1 - build added value services on top of IaaS and PaaS infrastructures
#2 - lower the entry barrier for non-skilled scientists



Which services are available?



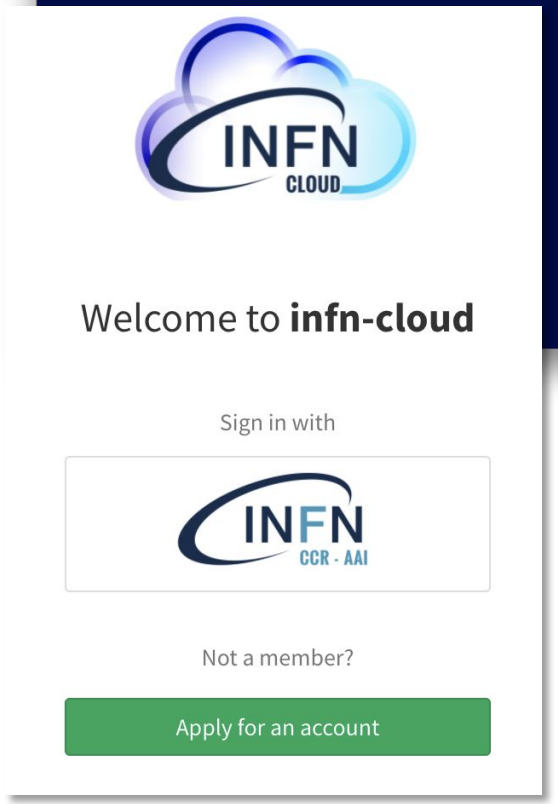
- Creation of VMs with different flavors and sizes.
- Creation of containers or of services via docker- compose files.
- Building blocks “as a service” for example for container orchestration (e.g. creation of a Mesos cluster or of a Kubernetes cluster as a service).
- Pre-configured environments for data analytics (e.g. using ElasticSearch and Kibana or Spark).
- Non volatile, object storage and Posix-compliant virtual file system solutions transparently connected to higher-layer services (e.g Jupyter notebooks as a service with permanent, replicated storage).
- Dynamic clusters tailored to specific experiments (e.g. an automated full HTCondor installation realized on a k8s cluster, or a GPU-based Machine Learning-optimized environment).
- Services leveraging transparent user-level encryption of disk volumes.

The service catalogue can be easily extended with the simple addition/customization of TOSCA templates.

The INFN Cloud Dashboard

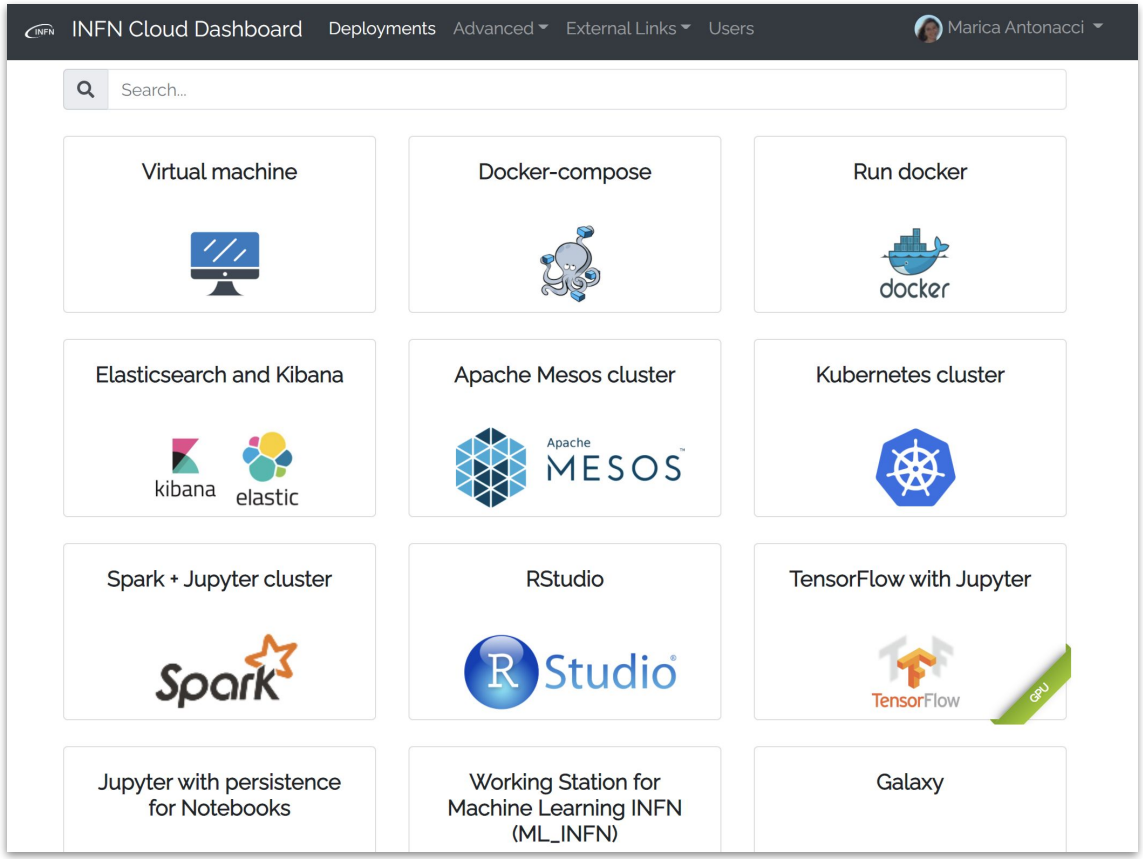
https://my.cloud.infn.it

INDIGO IAM manages the authentication/authorization through the whole stack (from PaaS to Iaas)



Users are organized in different IAM **groups**.

Each group can access a specific set of services from the dashboard (personalized view) and is mapped onto a dedicated tenant on the federated clouds.



The INFN Cloud Dashboard

Virtual machine

Docker-compose

Run docker

Elasticsearch and Kibana

Kubernetes cluster

Spark + Jupyter cluster

HTCondor cluster

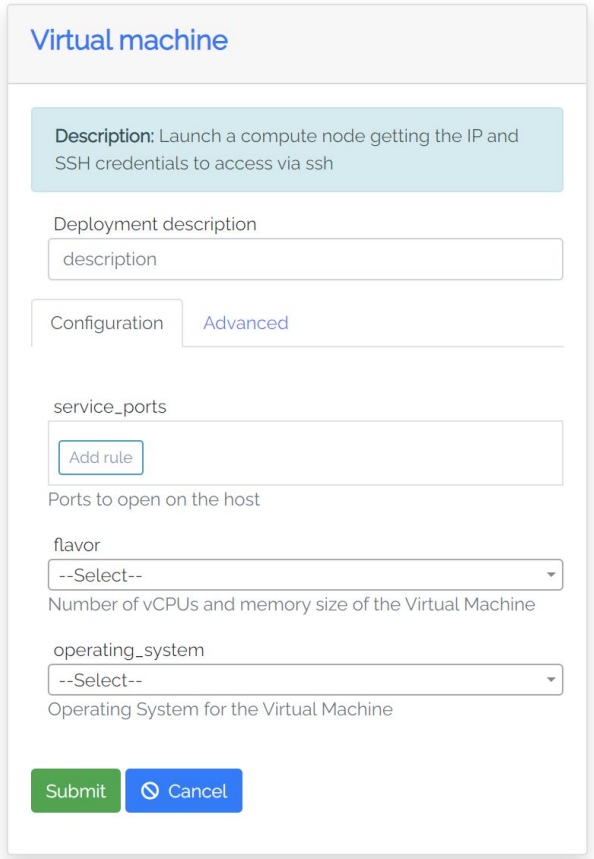
Jupyter with persistence for Notebooks

Computational enviroment for Machine Learning INFN (ML_INFN)

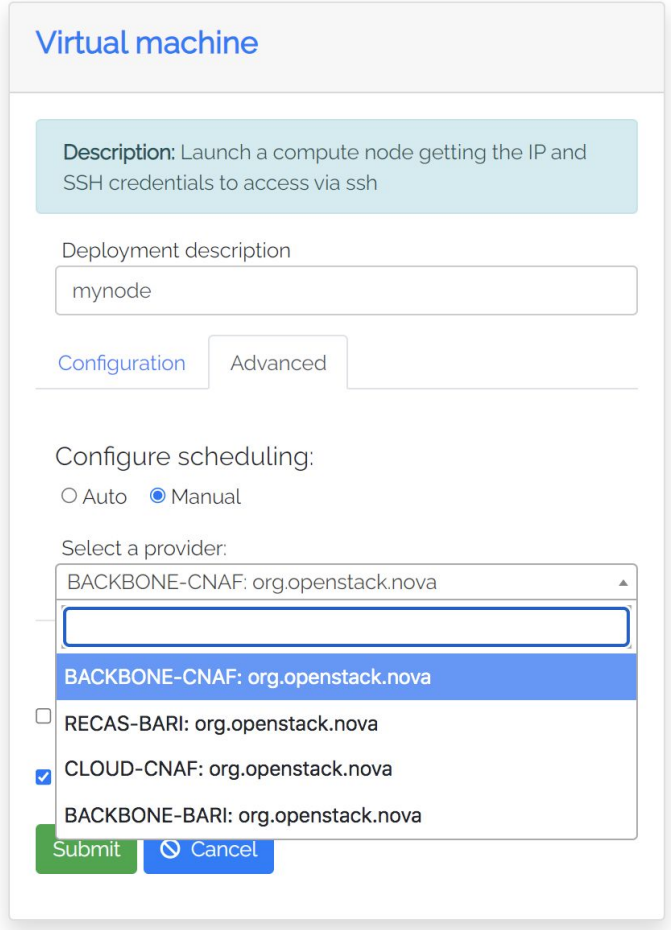
Working Station for CYGNO experiment

Sync&Share aaS

The services are **easily customizable** and configurable directly by users

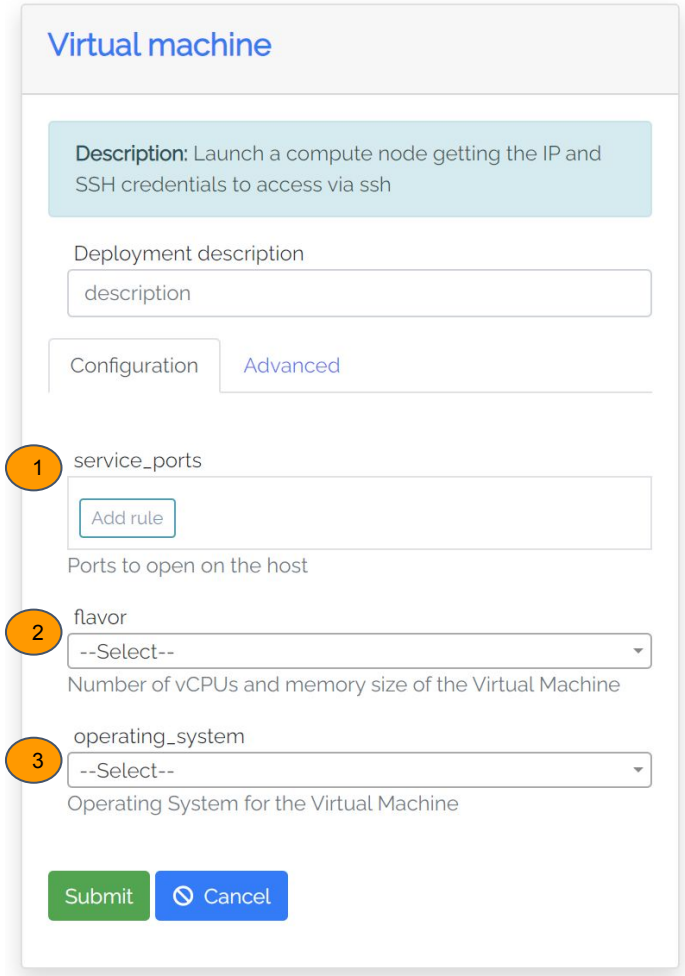


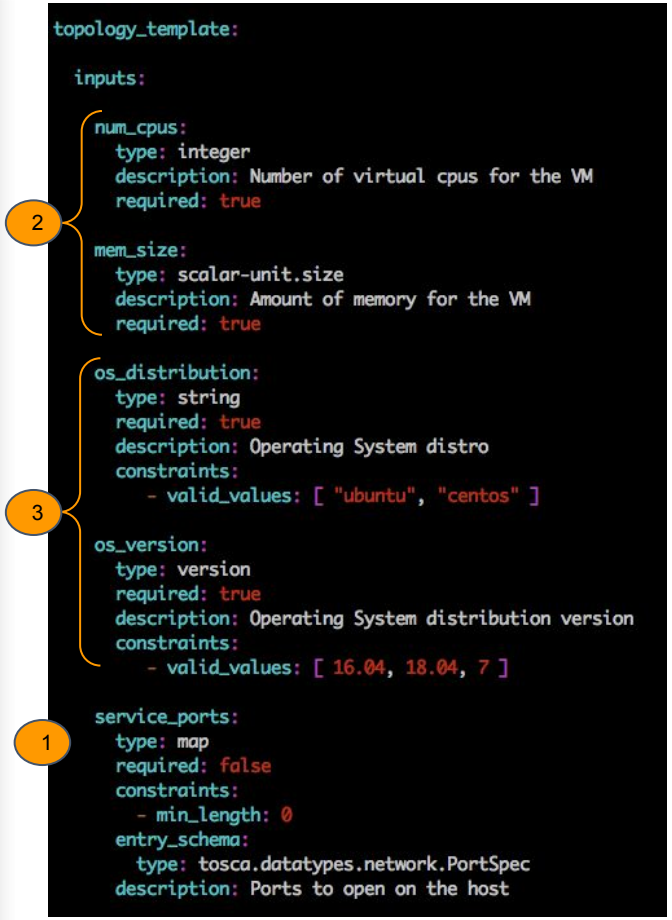
Transparent, multi-site **federation or site selection** made manually by the user



Marica Antonacci (marica.antonacci@ba.infn.it)

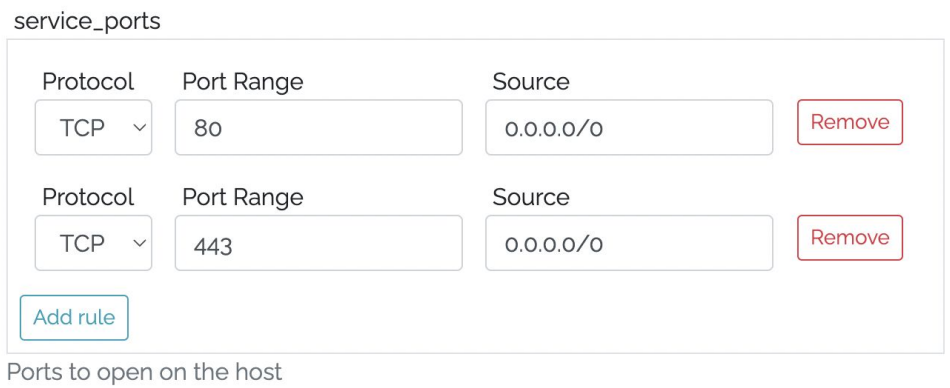
Service request customization



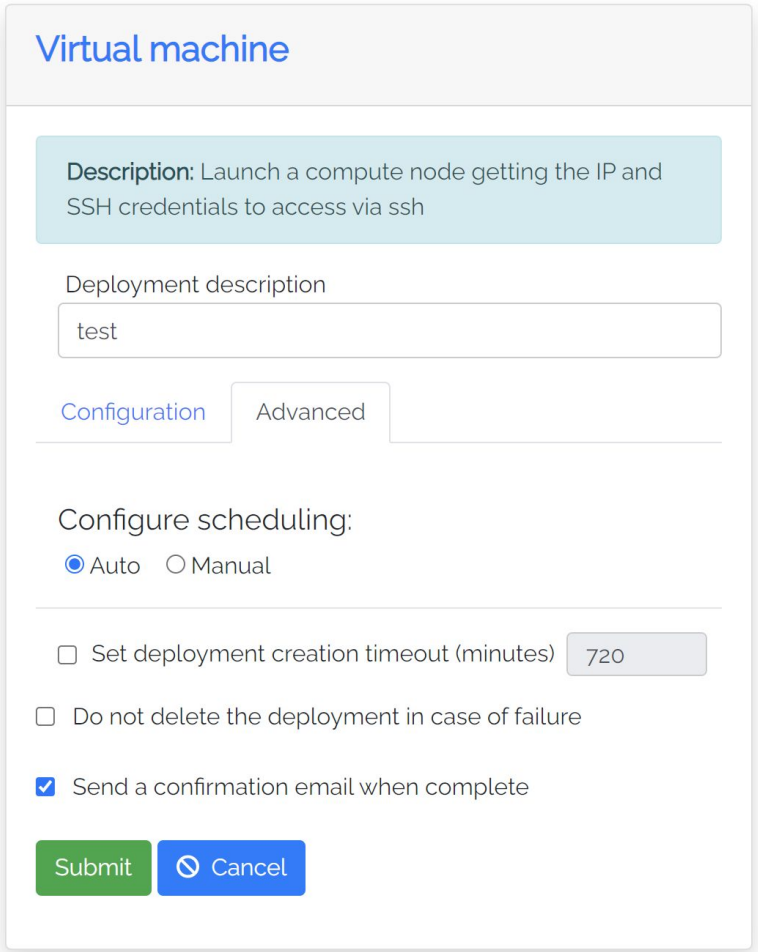


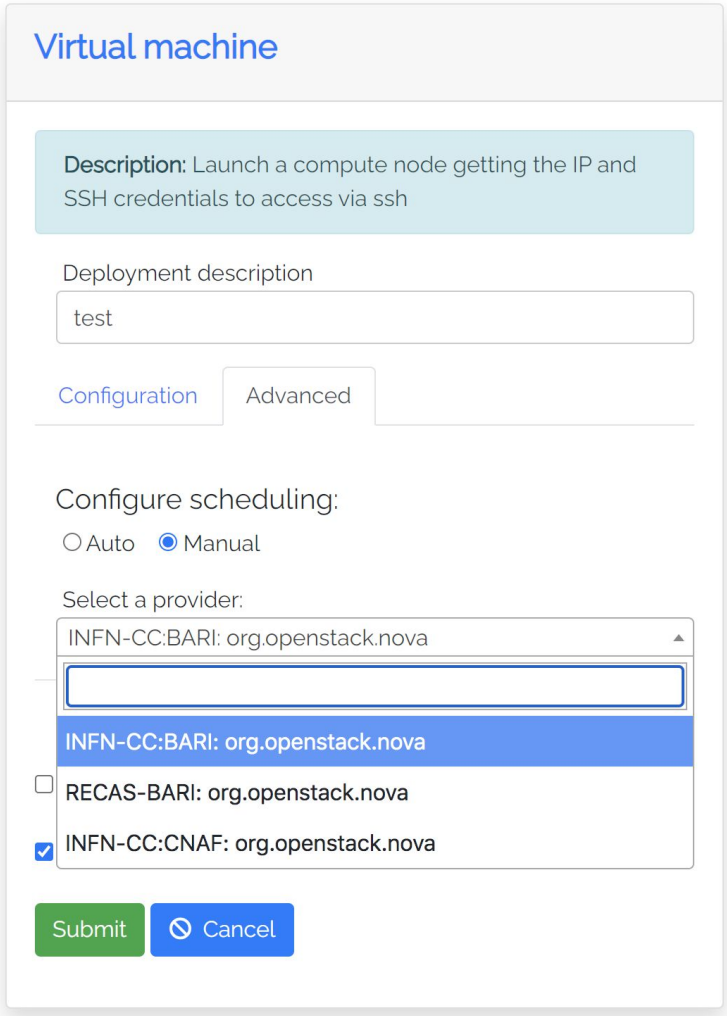
The configuration form allows the user to specify requirements for the deployment in a straightforward way

- checking the mandatory fields
- hiding the complexity of TOSCA
  - related fields are collapsed into a single input (e.g. num_cpu & mem_size into flavor)
  - complex TOSCA types are managed with dedicated Javascript functions (e.g.



Advanced configurations





The dashboard allows also to bypass the automatic scheduling implemented by the Orchestrator: the user can choose a specific provider to send his/her deployment request to.

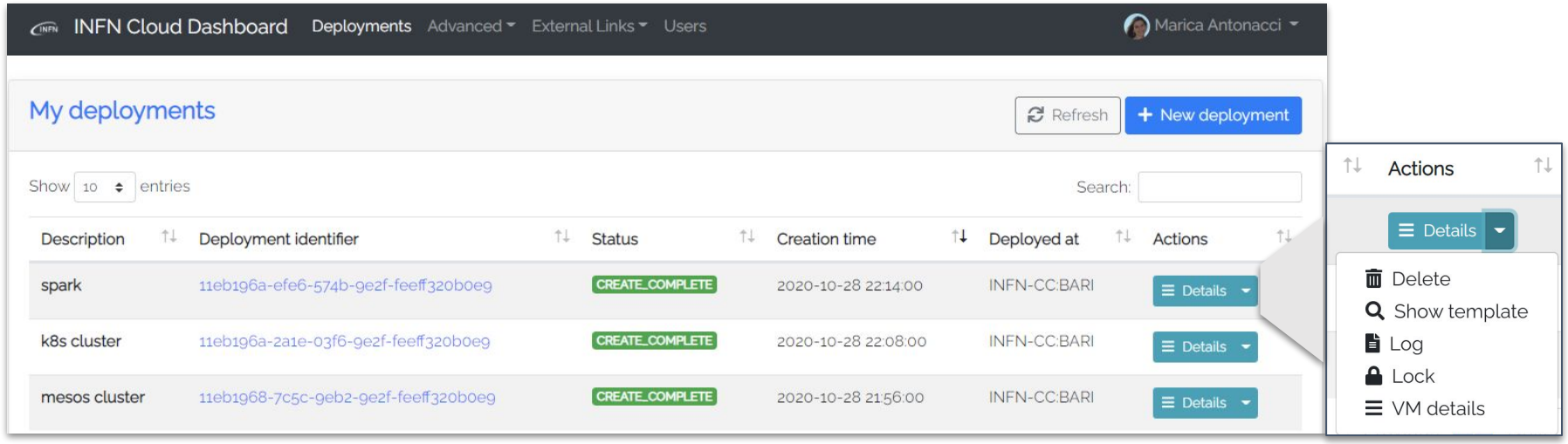
Under the hood:

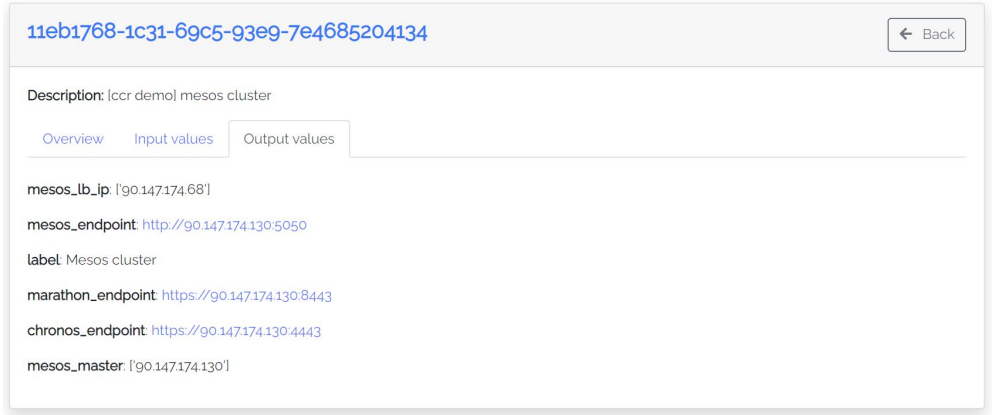
the drop-down menu is automatically created by the Dashboard interacting the SLA Manager Service to get the list of providers for the user;

before submitting the request to the Orchestrator, the Dashboard completes the TOSCA template including the proper SLA placement policy:

```
policies:
  - deploy_on_specific_site:
      type: tosca.policies.indigo.SlaPlacement
      properties:
        sla_id: 5e1daa90d000a819fe11ca56
```

Deployment outputs and notifications

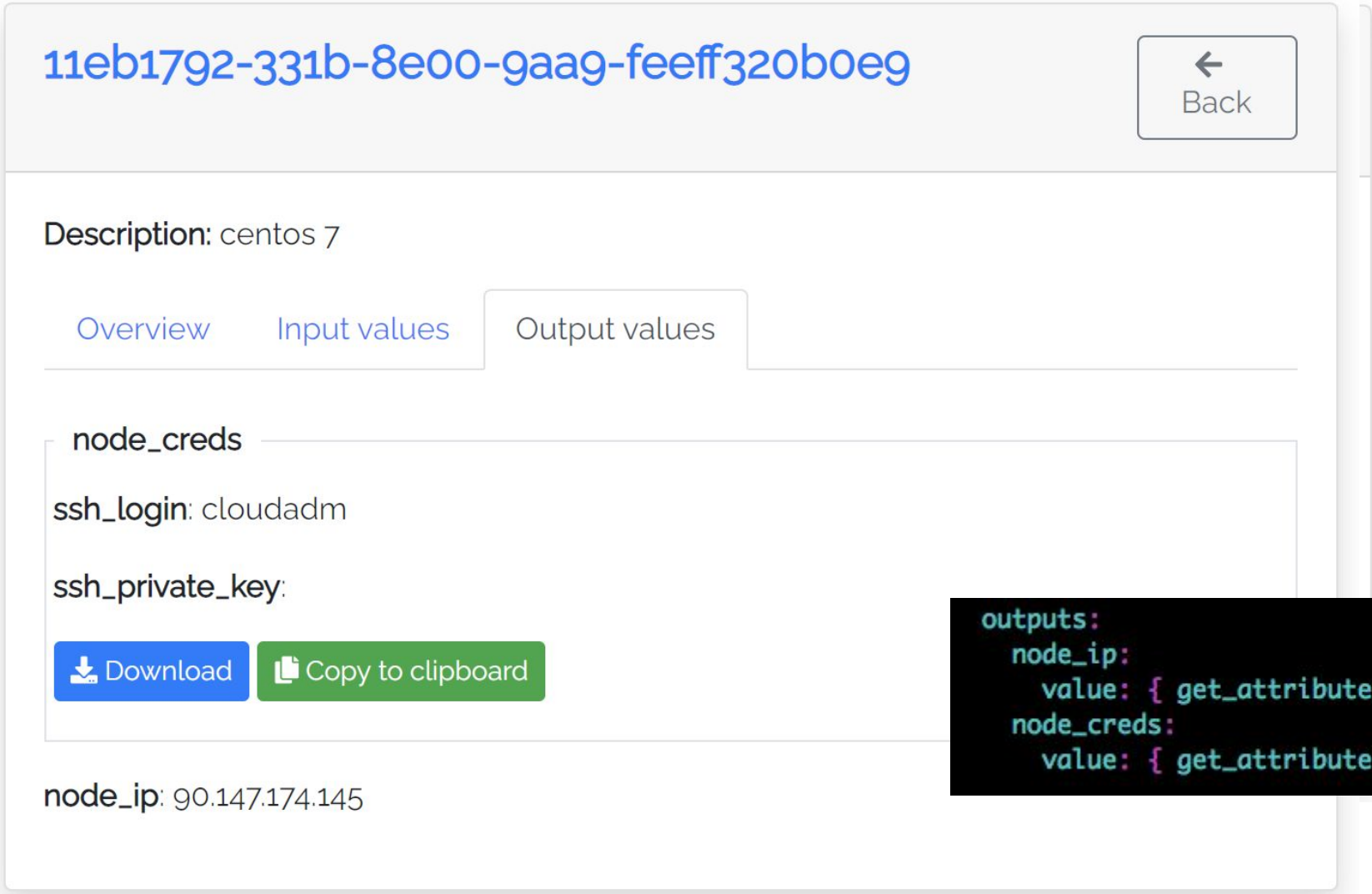




A notification system is implemented in the Dashboard: the user receives an automatic email as soon as the deployment is ready.

Then, the details about the deployed service can be accessed through the Dashboard.

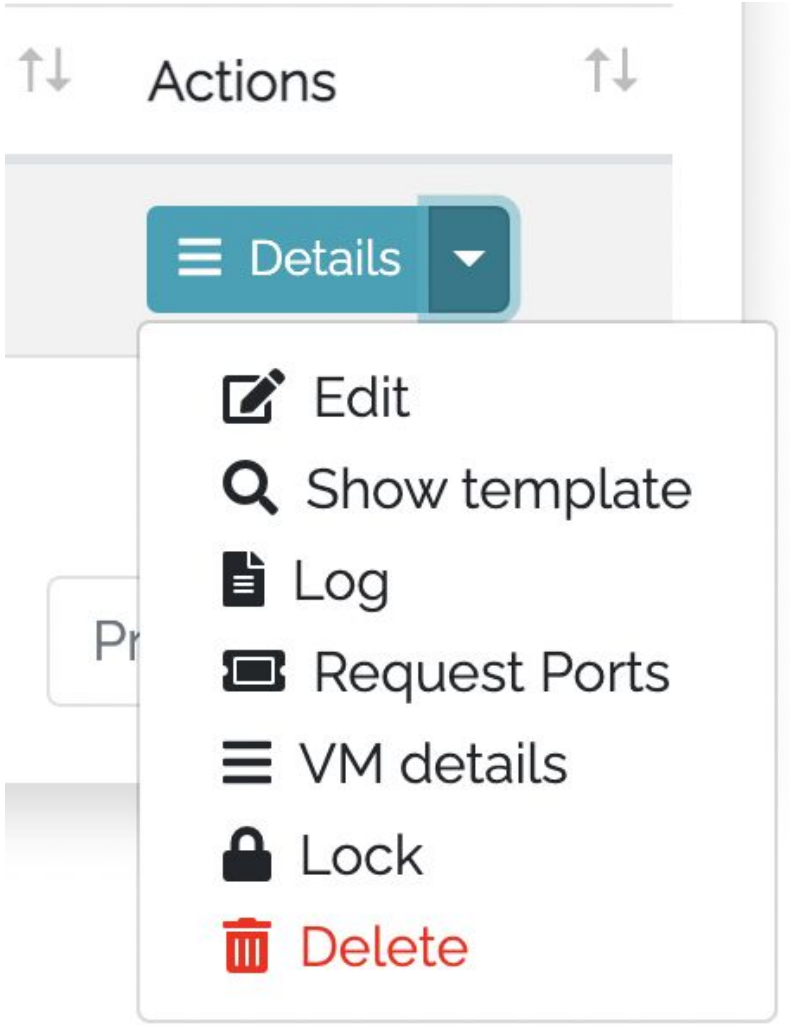
Deployment details



The outputs are defined in the tosca template of the service and are valuated at runtime

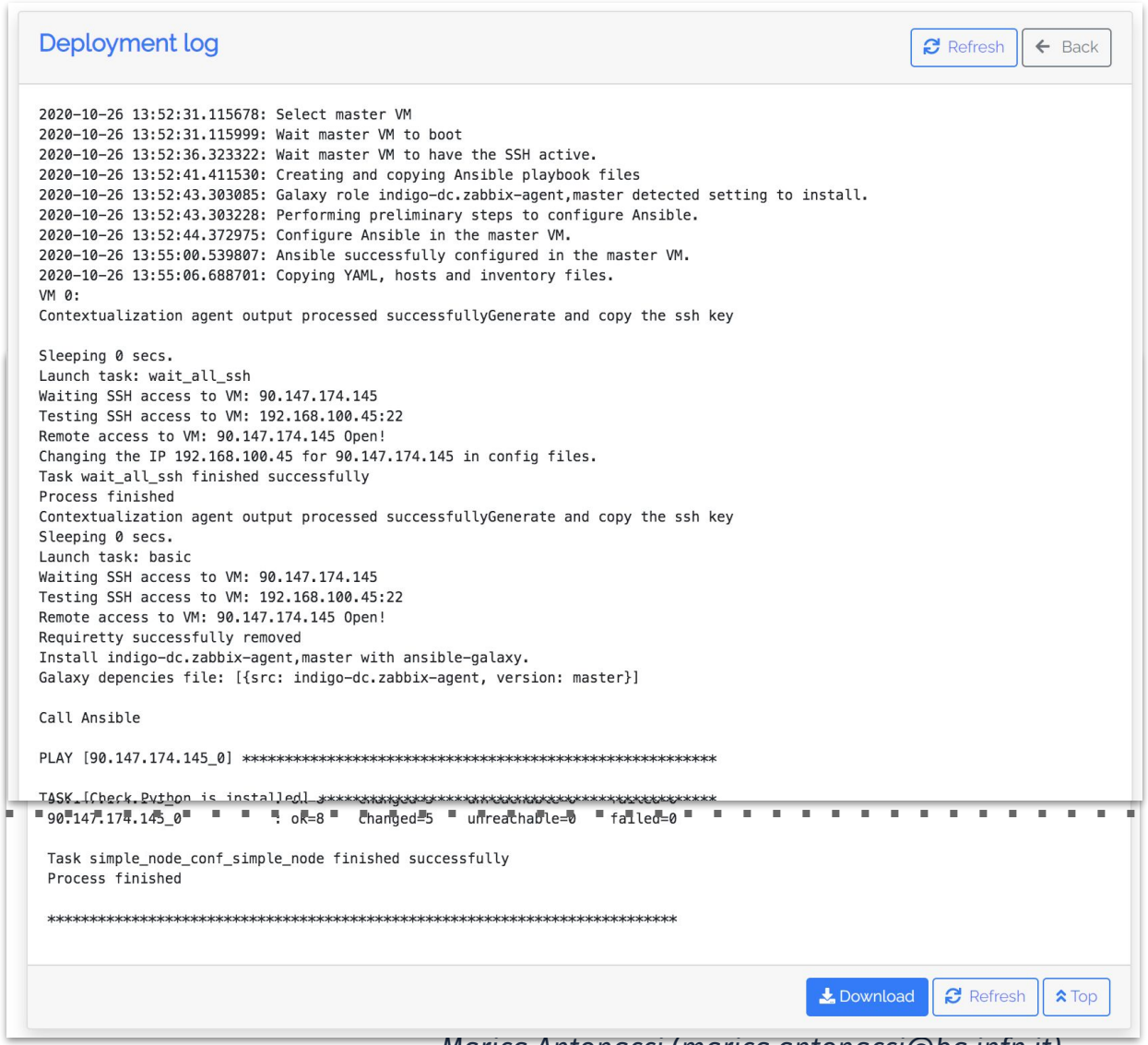
```
outputs:
  node_ip:
    value: { get_attribute: [ simple_node, public_address, 0 ] }
  node_creds:
    value: { get_attribute: [ simple_node, endpoint, credential, 0 ] }
```

Menu “Actions”

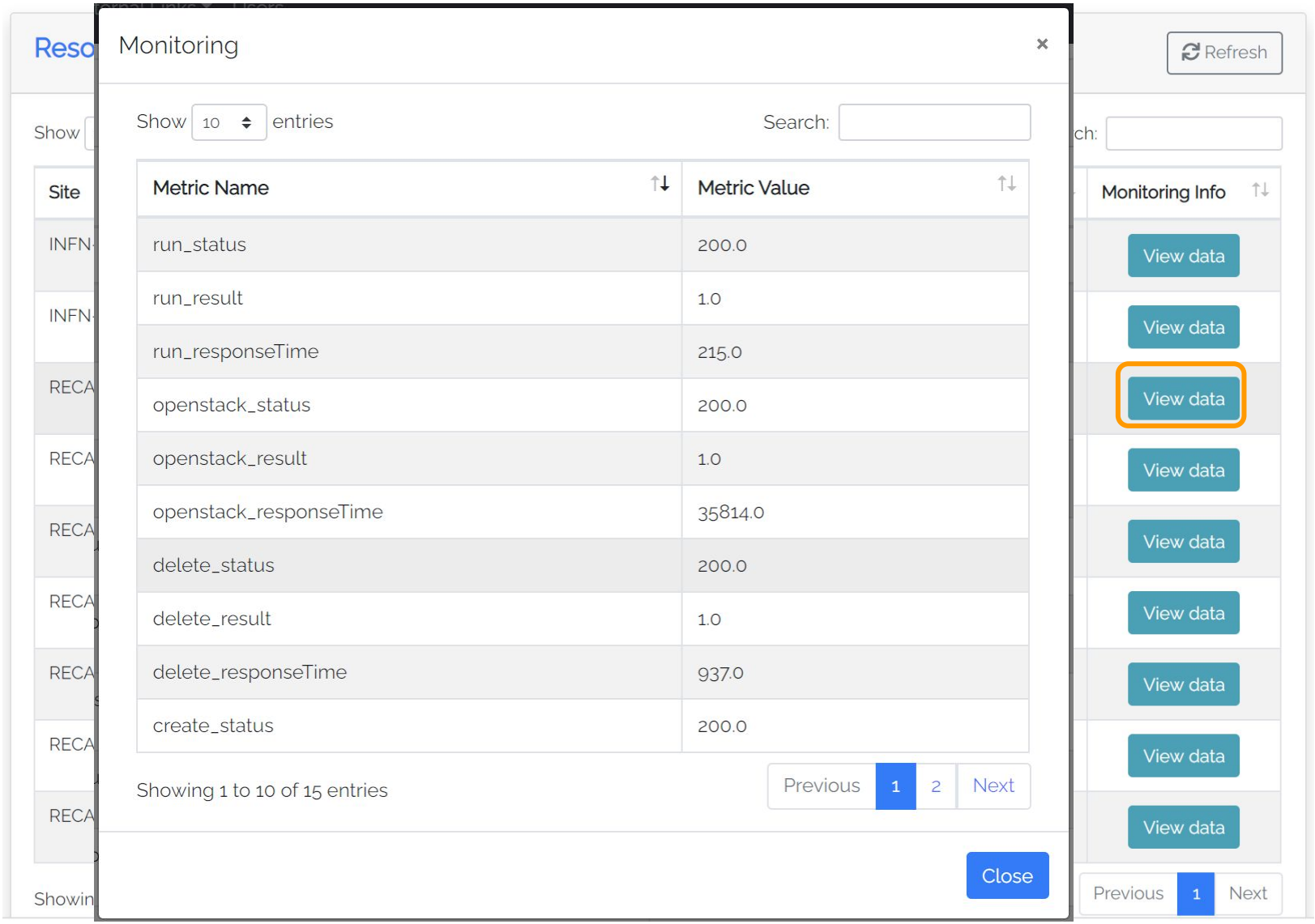


- **Delete**: remove the whole deployment
- **Show template**: view the TOSCA template used to make the deployment
- **Log**: view the contextualization log (generated by the Infrastructure Manager)
- **Lock**: protect deployment against delete operations
- **VM details** (only admin): get detailed information about the VMs of the deployment
- **Request Ports**: open automatically a ticket for the Support team to request a modification of the security group rules

Deployment log

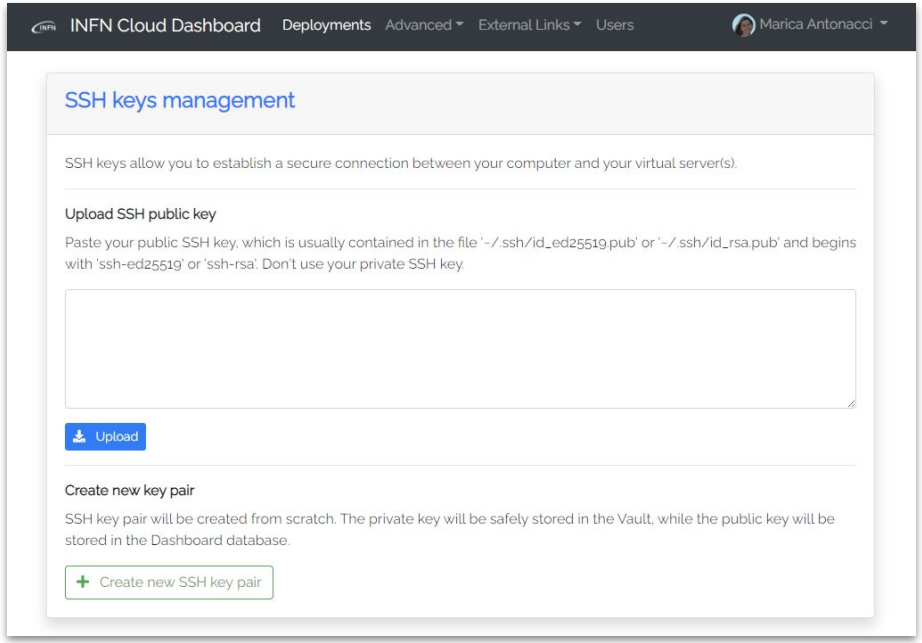


Resource Providers view (Advanced Menu)



The dashboard aggregates information retrieved from SLA manager, CMDB and Monitoring system.

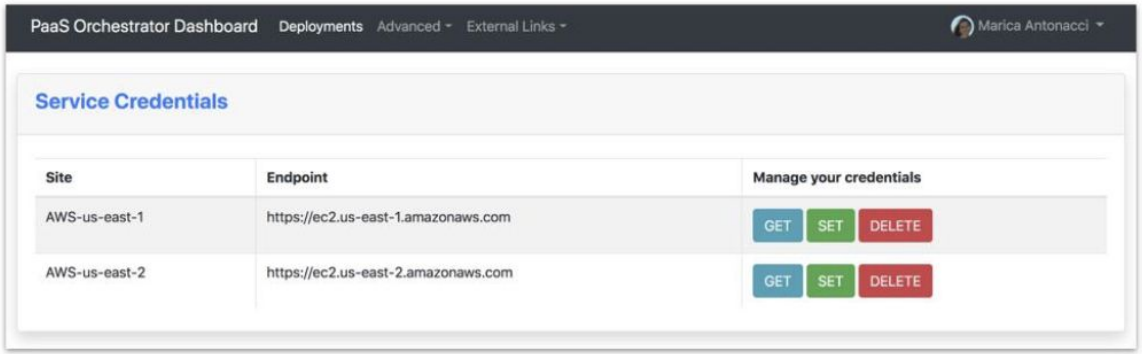
Secrets management

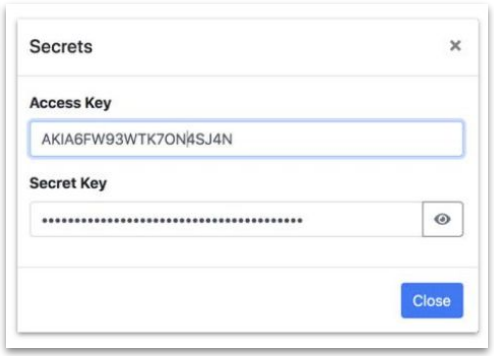


The Dashboard is integrated with Hashicorp **Vault** (Secrets Manager) to support some functionalities, e.g.

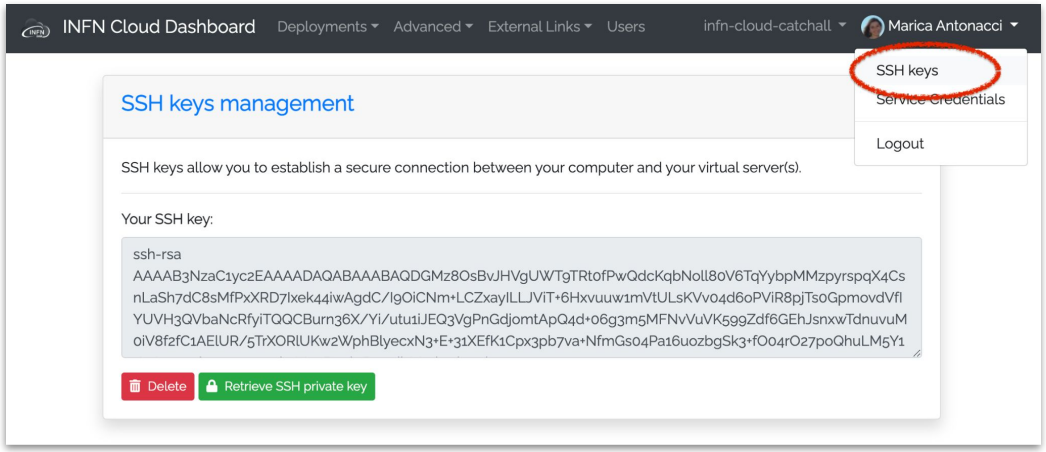
- **ssh key pair management**
- service credentials store (e.g. AWS)

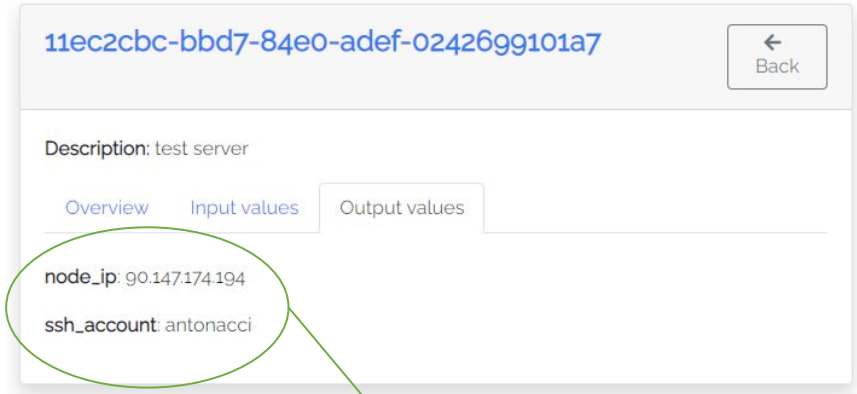
The Vault has been integrated with **INFN Cloud IAM** (jwt auth) and proper policies grant read and/or write permissions to specific Vault paths depending on the user claims.

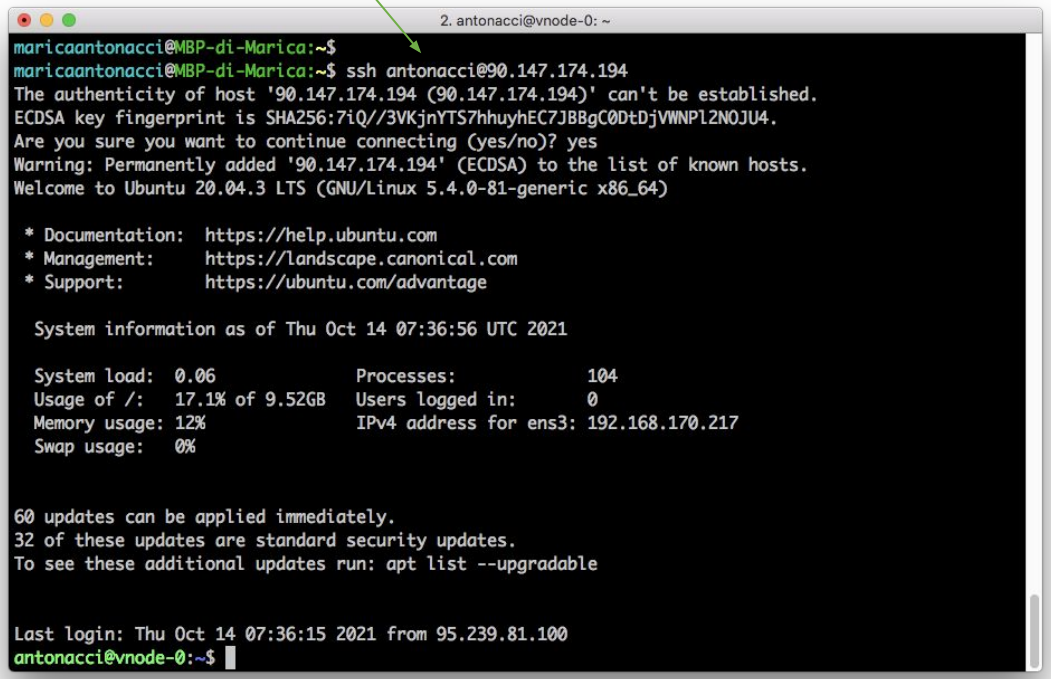




Access all your VMs with your username and ssh key







Conclusions

- The INFN Cloud PaaS Dashboard makes it easy to discover, select, configure and request the deployment of services that fit the needs and requirements of the INFN research communities.
- New applications and services are continuously included in the catalogue and the Dashboard is enriched with new functionalities to support them.
- Both the addition of a new service in the marketplace and the federation of a new resource provider are quite simple processes, thanks to the flexibility and extensibility of the PaaS architecture and implementation.

Thank you

for your attention!