



INFN Cloud Dashboard

Marica Antonacci (INFN BA)

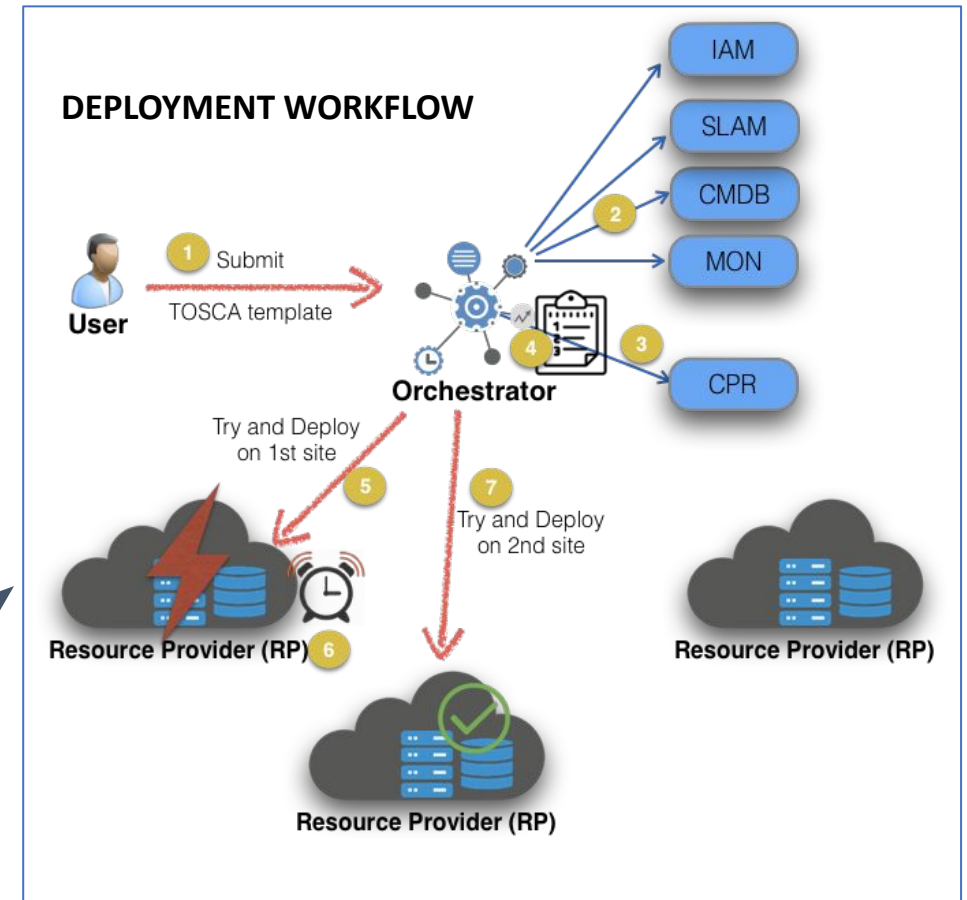
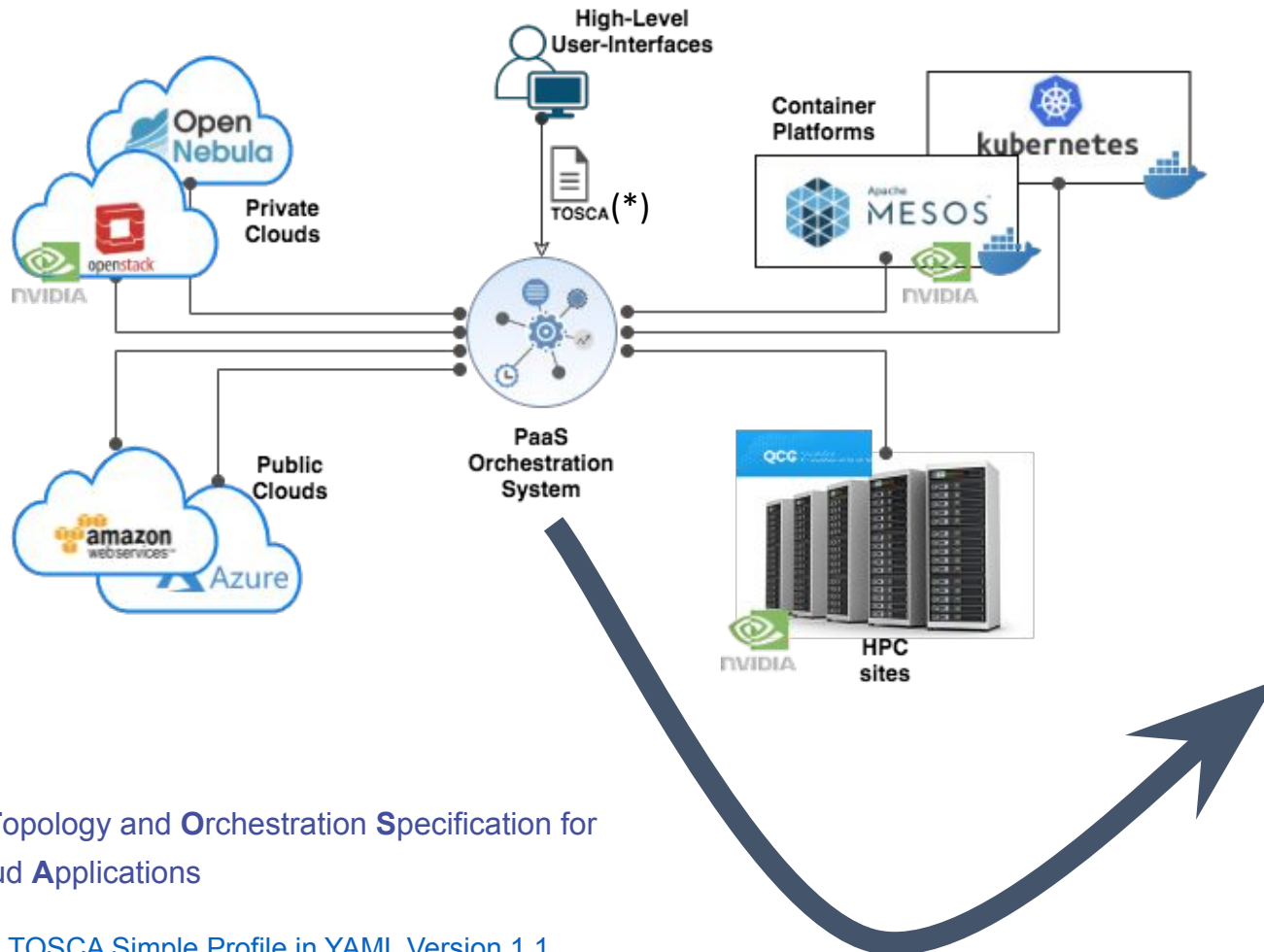
**Uso e sviluppo di applicazioni e servizi su INFN Cloud
(CLueApp)
13-16 September 2022**

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)



(*) Topology and Orchestration Specification for Cloud Applications

Ref: [TOSCA Simple Profile in YAML Version 1.1](#)

The INFN Cloud services



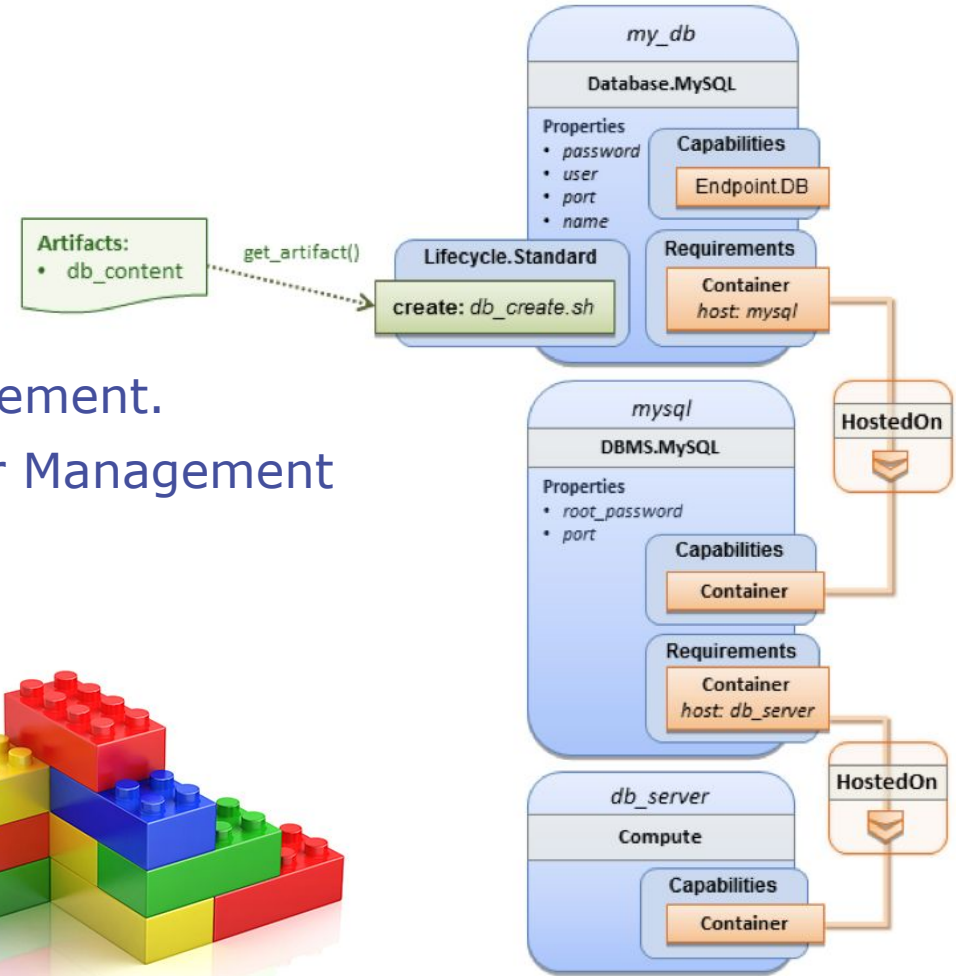
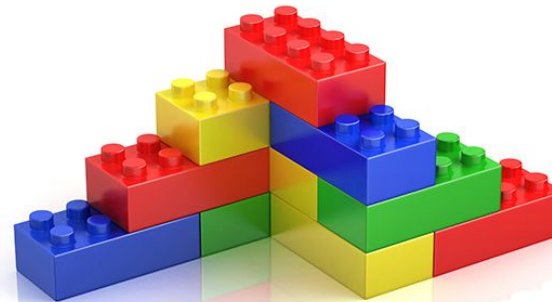
- The INFN Cloud services are based on **modular components and span the IaaS, PaaS and SaaS models** for both computing and data.
- All services are described by **TOSCA templates** (which can refer internally to other components such as Ansible playbooks, HELM charts, etc.).
- The services can be **deployed** 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 Orchestration Specification for Cloud Applications

- Goals:

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



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

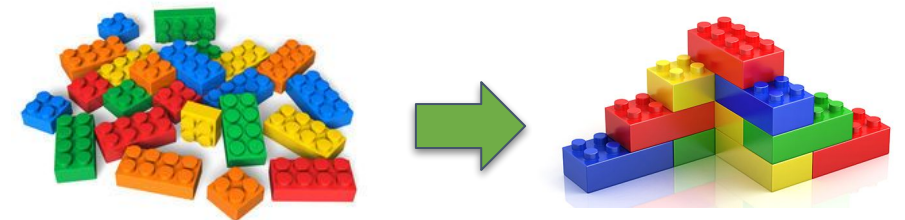


The catalogue is a graphical representation of the TOSCA templates repository that we have been developing extending the INDIGO-DC custom types

- 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?

SIMPLE

- 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.

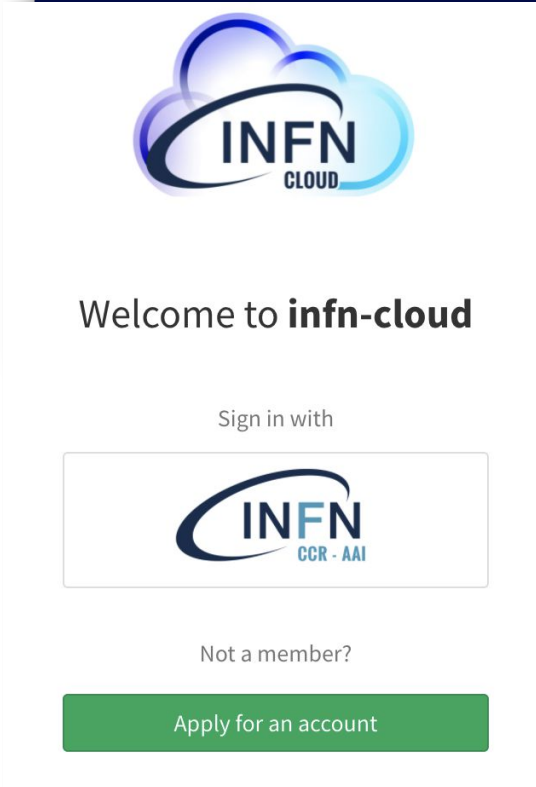
COMPLEX

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

The INFN Cloud Dashboard

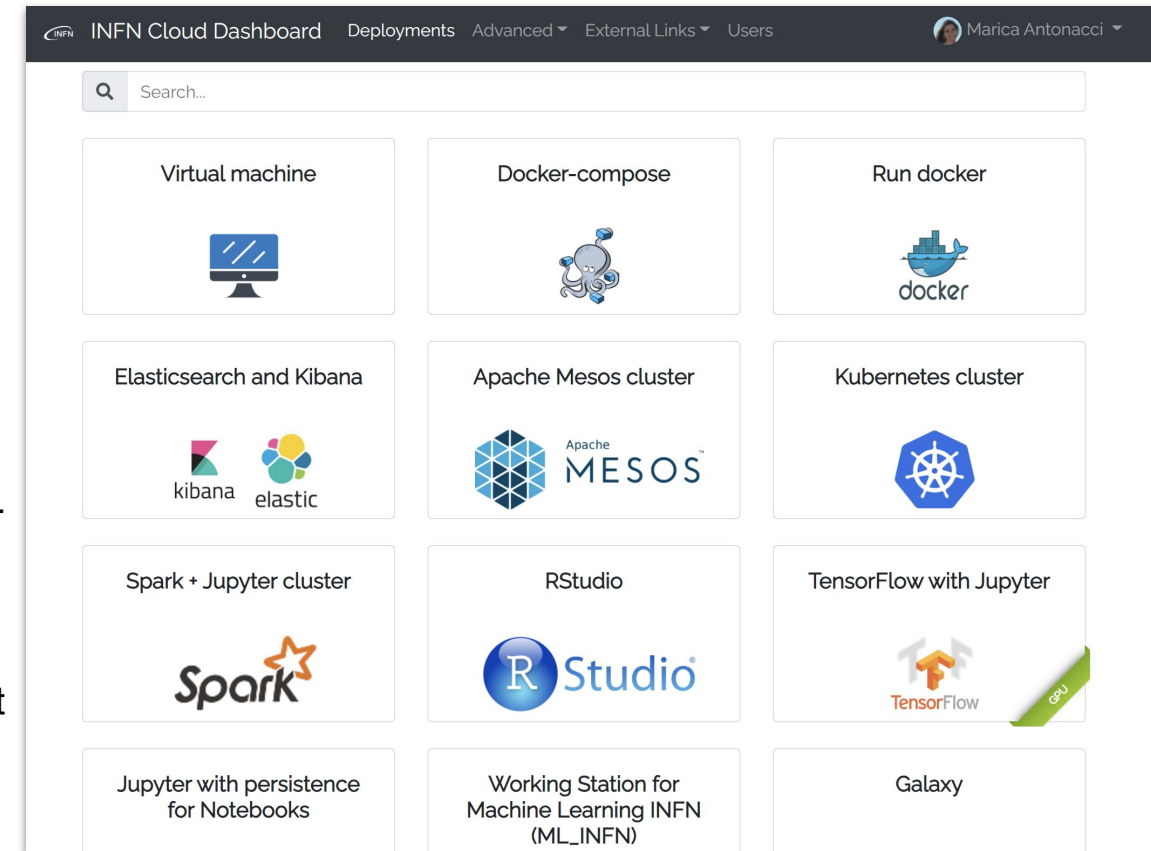


INDIGO IAM manages the authentication/authorization through the whole stack (from PaaS to IaaS)



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



<p>Virtual machine</p>	<p>Docker-compose</p>	<p>Run docker</p>
<p>Elasticsearch and Kibana</p>	<p>Kubernetes cluster</p>	<p>Spark + Jupyter cluster</p>
<p>HTCondor cluster</p>	<p>Jupyter with persistence for Notebooks</p>	<p>Computational environment for Machine Learning INFN (ML-INFN)</p>
<p>Working Station for CYGNO experiment</p>	<p>Sync&Share aaS</p>	

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

Virtual machine

Description: Launch a compute node getting the IP and SSH credentials to access via ssh

Deployment description
description

Configuration **Advanced**

service_ports

Ports to open on the host

flavor
--Select--

Number of vCPUs and memory size of the Virtual Machine

operating_system
--Select--

Operating System for the Virtual Machine

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

Virtual machine

Description: Launch a compute node getting the IP and SSH credentials to access via ssh

Deployment description
mynode

Configuration **Advanced**

Configure scheduling:
 Auto Manual

Select a provider:

- BACKBONE-CNAF: org.openstack.nova
- BACKBONE-CNAF: org.openstack.nova
- RECAS-BARI: org.openstack.nova
- CLOUD-CNAF: org.openstack.nova
- BACKBONE-BARI: org.openstack.nova

Service request customization

Virtual machine

Description: Launch a compute node getting the IP and SSH credentials to access via ssh

Deployment description

Configuration **Advanced**

1 service_ports

Ports to open on the host

2 flavor

Number of vCPUs and memory size of the Virtual Machine

3 operating_system

Operating System for the Virtual Machine

```
topology_template:
  inputs:
    num_cpus:
      type: integer
      description: Number of virtual cpus for the VM
      required: true
    mem_size:
      type: scalar-unit.size
      description: Amount of memory for the VM
      required: true
    os_distribution:
      type: string
      required: true
      description: Operating System distro
      constraints:
        - valid_values: [ "ubuntu", "centos" ]
    os_version:
      type: version
      required: true
      description: Operating System distribution version
      constraints:
        - valid_values: [ "16.04", "18.04", "7" ]
    service_ports:
      type: map
      required: false
      constraints:
        - min_length: 0
      entry_schema:
        type: tosca.datatypes.network.PortSpec
        description: Ports to open on the host
```

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.

service_ports

Protocol	Port Range	Source	
TCP	80	0.0.0.0/0	<input type="button" value="Remove"/>
TCP	443	0.0.0.0/0	<input type="button" value="Remove"/>

Ports to open on the host

Advanced configurations

Virtual machine

Description: Launch a compute node getting the IP and SSH credentials to access via ssh

Deployment description
test

Configuration **Advanced**

Configure scheduling:
 Auto Manual

Set deployment creation timeout (minutes) 720

Do not delete the deployment in case of failure

Send a confirmation email when complete

Submit **Cancel**

Virtual machine

Description: Launch a compute node getting the IP and SSH credentials to access via ssh

Deployment description
test

Configuration **Advanced**

Configure scheduling:
 Auto Manual

Select a provider:

INFN-CC:BARI: org.openstack.nova

INFN-CC:BARI: org.openstack.nova

RECAS-BARI: org.openstack.nova

INFN-CC:CNAF: org.openstack.nova

Submit **Cancel**

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: toasca.policies.indigo.SlaPlacement  
  properties:  
    sla_id: 5e1daa90d000a819fe11ca56
```

Deployment outputs and notifications

The screenshot shows the 'My deployments' section of the INFN Cloud Dashboard. It features a table with columns for Description, Deployment identifier, Status, Creation time, Deployed at, and Actions. Three deployments are listed: 'spark', 'k8s cluster', and 'mesos cluster', all with a 'CREATE_COMPLETE' status. An 'Actions' dropdown menu is open for the 'mesos cluster' deployment, showing options: Details, Delete, Show template, Log, Lock, and VM details.

Description	Deployment identifier	Status	Creation time	Deployed at	Actions
spark	11eb196a-efe6-574b-9e2f-feeff320b0e9	CREATE_COMPLETE	2020-10-28 22:14:00	INFN-CC-BARI	Details
k8s cluster	11eb196a-2a1e-03f6-9e2f-feeff320b0e9	CREATE_COMPLETE	2020-10-28 22:08:00	INFN-CC-BARI	Details
mesos cluster	11eb1968-7c5c-9eb2-9e2f-feeff320b0e9	CREATE_COMPLETE	2020-10-28 21:56:00	INFN-CC-BARI	Details

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.

The screenshot shows an email notification from 'mycloud@infn.it' with the subject 'Deployment complete'. The email body includes the INFN logo and the text: 'Dear User, This is an automatically generated notification mail YOU DO NOT NEED TO ANSWER THIS MESSAGE. Your deployment 11eb1792-331b-8e00-9aa9-feeff320b0e9 is complete. Kind Regards.'

The screenshot shows the details page for the deployment '11eb1768-1c31-69c5-93e9-7e4685204134'. It includes a description 'lccr demo mesos cluster' and tabs for 'Overview', 'Input values', and 'Output values'. The 'Output values' tab is active, showing details for 'mesos_lb_ip', 'mesos_endpoint', 'label', 'marathon_endpoint', 'chronos_endpoint', and 'mesos_master'.

Deployment details

11eb1792-331b-8e00-9aa9-fee320b0e9

←
Back

Description: centos 7

Overview

Input values

Output values

node_creds

ssh_login: cloudadm

ssh_private_key:

Download

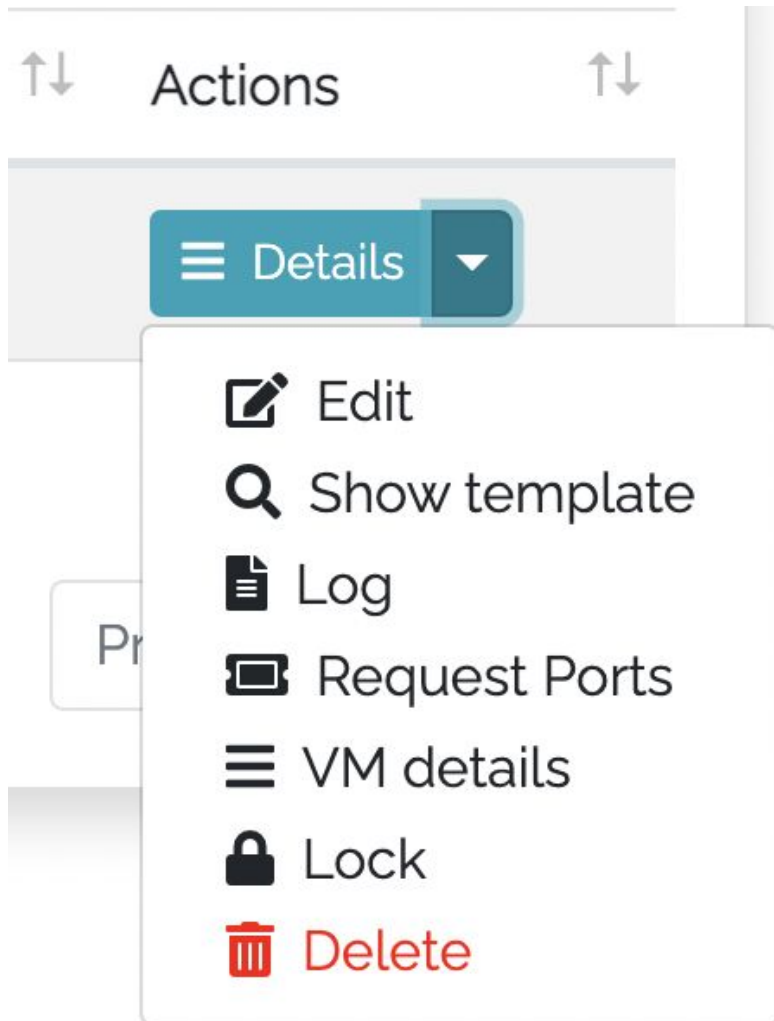
Copy to clipboard

node_ip: 90.147.174.145

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

Deployment log

Refresh Back

```
2020-10-26 13:52:31.115678: Select master VM
2020-10-26 13:52:31.115999: Wait master VM to boot
2020-10-26 13:52:36.323322: Wait master VM to have the SSH active.
2020-10-26 13:52:41.411530: Creating and copying Ansible playbook files
2020-10-26 13:52:43.303085: Galaxy role indigo-dc.zabbix-agent,master detected setting to install.
2020-10-26 13:52:43.303228: Performing preliminary steps to configure Ansible.
2020-10-26 13:52:44.372975: Configure Ansible in the master VM.
2020-10-26 13:55:00.539807: Ansible successfully configured in the master VM.
2020-10-26 13:55:06.688701: Copying YAML, hosts and inventory files.
VM 0:
Contextualization agent output processed successfullyGenerate and copy the ssh key

Sleeping 0 secs.
Launch task: wait_all_ssh
Waiting SSH access to VM: 90.147.174.145
Testing SSH access to VM: 192.168.100.45:22
Remote access to VM: 90.147.174.145 Open!
Changing the IP 192.168.100.45 for 90.147.174.145 in config files.
Task wait_all_ssh finished successfully
Process finished
Contextualization agent output processed successfullyGenerate and copy the ssh key
Sleeping 0 secs.
Launch task: basic
Waiting SSH access to VM: 90.147.174.145
Testing SSH access to VM: 192.168.100.45:22
Remote access to VM: 90.147.174.145 Open!
Requiretty successfully removed
Install indigo-dc.zabbix-agent,master with ansible-galaxy.
Galaxy dependencies file: [{src: indigo-dc.zabbix-agent, version: master}]

Call Ansible

PLAY [90.147.174.145_0] *****

TASK [Check Python is installed] *****
90.147.174.145_0 | oK=8 | Changed=5 | unreachable=0 | failed=0

Task simple_node_conf_simple_node finished successfully
Process finished

*****
```

Download Refresh Top

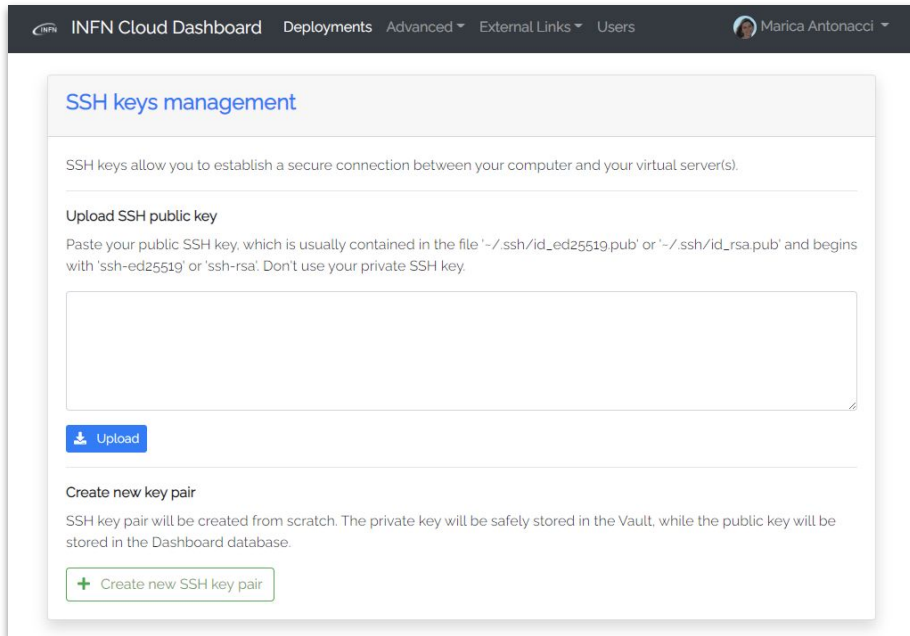
Resource Providers view (Advanced Menu)

The screenshot displays a 'Monitoring' window with a table of metrics. The table has two columns: 'Metric Name' and 'Metric Value'. The metrics listed are run_status, run_result, run_responseTime, openstack_status, openstack_result, openstack_responseTime, delete_status, delete_result, delete_responseTime, and create_status. The values range from 1.0 to 35814.0. A 'View data' button is highlighted with an orange box in the right-hand panel of the monitoring interface.

Metric Name	Metric Value
run_status	200.0
run_result	1.0
run_responseTime	215.0
openstack_status	200.0
openstack_result	1.0
openstack_responseTime	35814.0
delete_status	200.0
delete_result	1.0
delete_responseTime	937.0
create_status	200.0

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

Secrets management



INFN Cloud Dashboard | Deployments | Advanced | External Links | Users | Marica Antonacci

SSH keys management

SSH keys allow you to establish a secure connection between your computer and your virtual server(s).

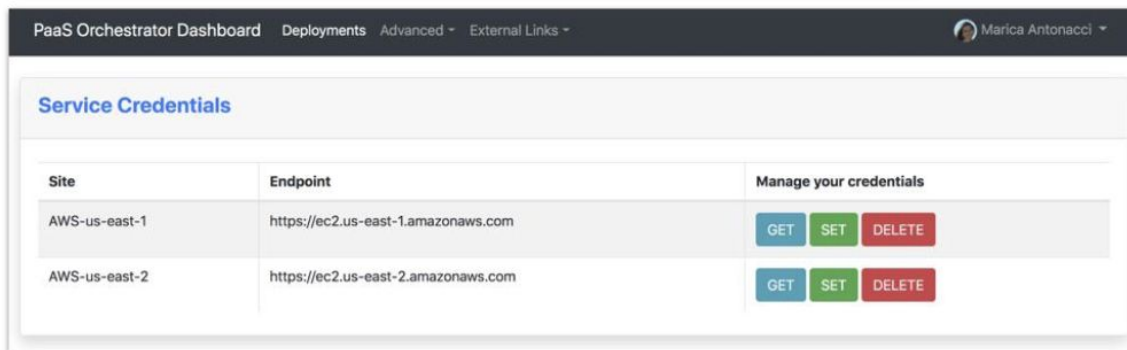
Upload SSH public key
Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Don't use your private SSH key.

Create new key pair
SSH key pair will be created from scratch. The private key will be safely stored in the Vault, while the public key will be stored in the Dashboard database.

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)

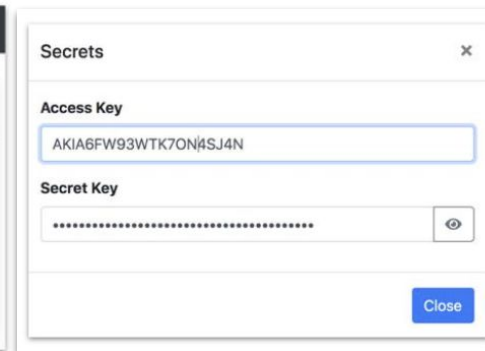
The Vault has been integrated with **INFN Cloud IAM** (jwt auth) and proper policies grant read and/or write permissions to specific Vault paths depending on the user claims.



PaaS Orchestrator Dashboard | Deployments | Advanced | External Links | Marica Antonacci

Service Credentials

Site	Endpoint	Manage your credentials
AWS-us-east-1	https://ec2.us-east-1.amazonaws.com	<input type="button" value="GET"/> <input type="button" value="SET"/> <input type="button" value="DELETE"/>
AWS-us-east-2	https://ec2.us-east-2.amazonaws.com	<input type="button" value="GET"/> <input type="button" value="SET"/> <input type="button" value="DELETE"/>



Secrets

Access Key

Secret Key

Access all your VMs with your username and ssh key



INFN Cloud Dashboard

SSH keys management

SSH keys allow you to establish a secure connection between your computer and your virtual server(s).

Your SSH key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDAQMz80sBvJHVgUWTgTRtofPwQdcKqbNoll8oV6TqYybpMMzpyrspqX4Cs
nLaSh7dC8sMFPxXRD7lxek44wAgdC/IgOICNm+LCZxayILLJVIT+6Hxvuuw1mVtULsKVv04d6oPvIR8pjTsoGpmovdVfl
YUVH3QVbaNcRfyITQCCBum36X/Yi/utu1JJEQ3VgPnGdjomtApQ4d+06g3m5MFNVuVK599zdf6GEHJsxnwTdnvuuM
oIV8fzFC1AEIUR/gTrXORLUKw2WphBlyecN3+E+31Xefk1Cpx3pb7va+NfmGso4Pa16uozbgSk3+fOo4rO27poQhuLM5Y1
```

Delete Retrieve SSH private key

11ec2cbc-bbd7-84e0-ade0-0242699101a7

Back

Description: test server

Overview Input values Output values

node_ip: 90.147.174.194

ssh_account: antonacci

```
2. antonacci@vnode-0: ~
maricaantonacci@MBP-di-Marica:~$
maricaantonacci@MBP-di-Marica:~$ ssh antonacci@90.147.174.194
The authenticity of host '90.147.174.194 (90.147.174.194)' can't be established.
ECDSA key fingerprint is SHA256:7iQ//3VKjnYTS7hhuyhEC7JBBgC0DtDjVWNPL2NOJU4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '90.147.174.194' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Thu Oct 14 07:36:56 UTC 2021

System load: 0.06          Processes:            104
Usage of /:   17.1% of 9.52GB  Users logged in:     0
Memory usage: 12%          IPv4 address for ens3: 192.168.170.217
Swap usage:  0%

60 updates can be applied immediately.
32 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Oct 14 07:36:15 2021 from 95.239.81.100
antonacci@vnode-0:~$
```

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!

