Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
terabit

Efficient management of INDIGO-IAM clients and S3 buckets via INDIGO PaaS Orchestrator in INFN Cloud

Luca Giommi – INFN CNAF

E. Vianello, R. Miccoli, F. Agostini, F. Fornari, A. Costantini – INFN CNAF

M. Antonacci, G. Vino, G. Savarese, G. Donvito – INFN Bari

Taipei (Taiwan) , 26/03/2024

International Symposium on Grids & Clouds (ISGC) 2024

INFN and its facilities

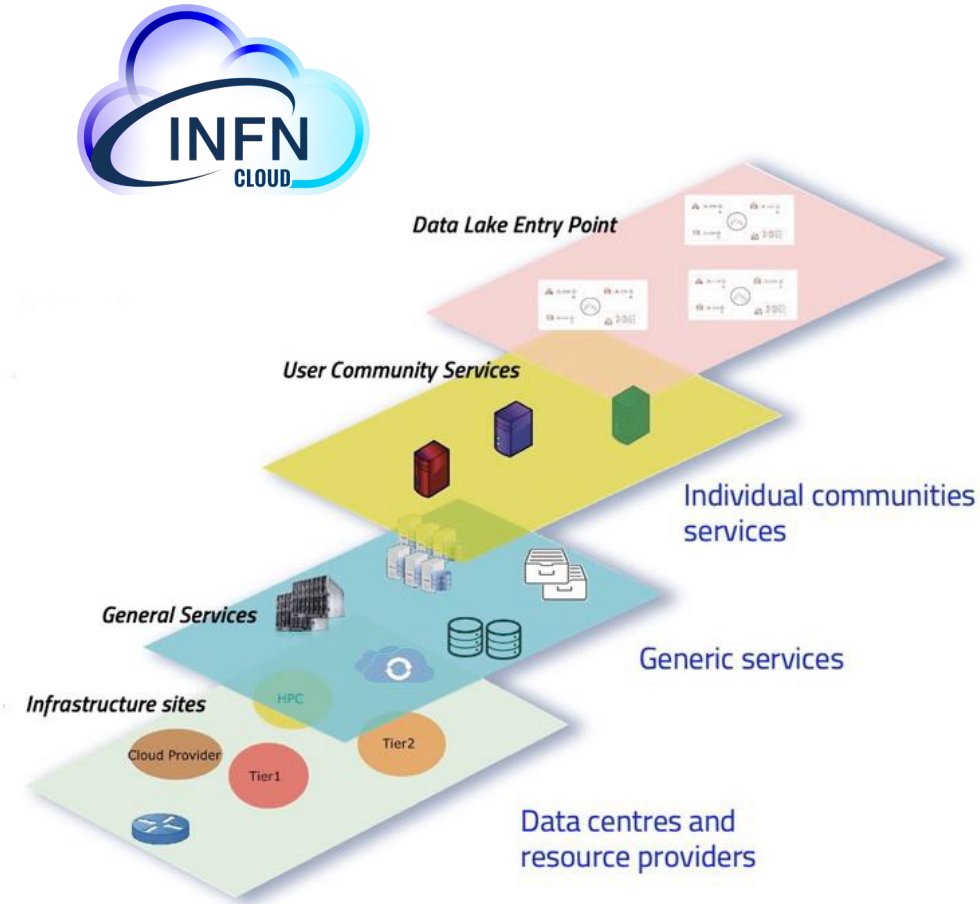
- INFN manages and supports the **largest public computing infrastructure for scientific research** spread throughout the country
- INFN has been running for more than 20 years a **distributed infrastructure** which currently offers about 150K CPU cores, 120 PB of enterprise-level disk space and 120 PB of tape storage, serving more than 40 international scientific collaborations
- INFN was one of main promoters of the GRID project to address LHC computing needs. Since then INFN has been participating to **WLCG** that includes more than 170 sites around the world, loosely organized in a tiered model.
  - In Italy, there are the Tier-1 at CNAF, Bologna and 9 Tier-2 centers

Birth of INFN Cloud

- To support and evolve use cases that could not easily exploit the Grid paradigm, for many years several INFN sites have been investing in **Cloud computing** infrastructures
  - heterogeneous in hardware, software and cloud middleware

- To optimize the use of available resources and expertise, INFN decided to implement a **national Cloud infrastructure** for research
  - as a **federation** of existing distributed infrastructures extending them if necessary in a transparent way to private and commercial providers
  - as an "user-centric" infrastructure making available to the final users a dynamic **set of services** tailored on specific use cases
  - leveraging the outcomes of several national and European cloud projects where INFN actively participated

- INFN Cloud was officially made available to users in **March 2021**



Proposed architecture for ICSC and TeRABIT projects

Resources in INFN CLoud

The infrastructure is based on a core **backbone** connecting the large data centers of CNAF and Bari, and on a set of loosely coupled distributed and federated sites connected to the backbone

- Backbone sites are high speed connected and host the INFN Cloud core services
- **Federated clouds**: Cloud@CNAF, CloudVeneto, Cloud@ReCaS-Bari, Cloud-CT, Cloud-IBISCO-Na. Coming soon: LNGS, Milano, HTC in Tier-2s, HPC bubbles

Backbone

~ 2000 vCPU
~ 15 TB RAM
~ 1.6 PB Storage (RAW)
> 600 TB Storage net, ~ 10% SSD, ~ 320 TB for object storage

Federated Clouds

~ 3160 vCPU
~ 75 TB RAM
~ 334 TB Storage net

Portfolio of services

SaaS

- Notebook as a Service
- INFN Cloud Registry (Harbor)
- INFN Cloud object storage (Minio)
- INFN Cloud monitoring (Grafana)

PaaS

- Virtual Machine
- Docker Compose
- Run Docker
- INDIGO IAM as a Service
- Elasticsearch & Kibana
- Kubernetes cluster
- Spark + Jupyter cluster
- HTCondor (mini or cluster)
- Jupyter (w/o Matlab) with persistence
- Sync & Share
- ML_INFN working station
- CYGNO working station

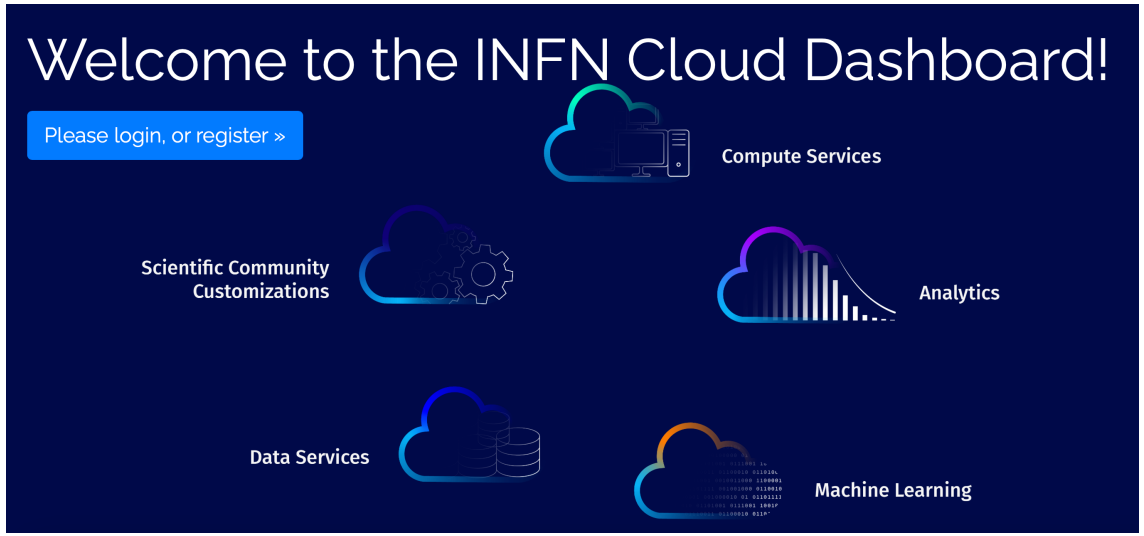
IaaS

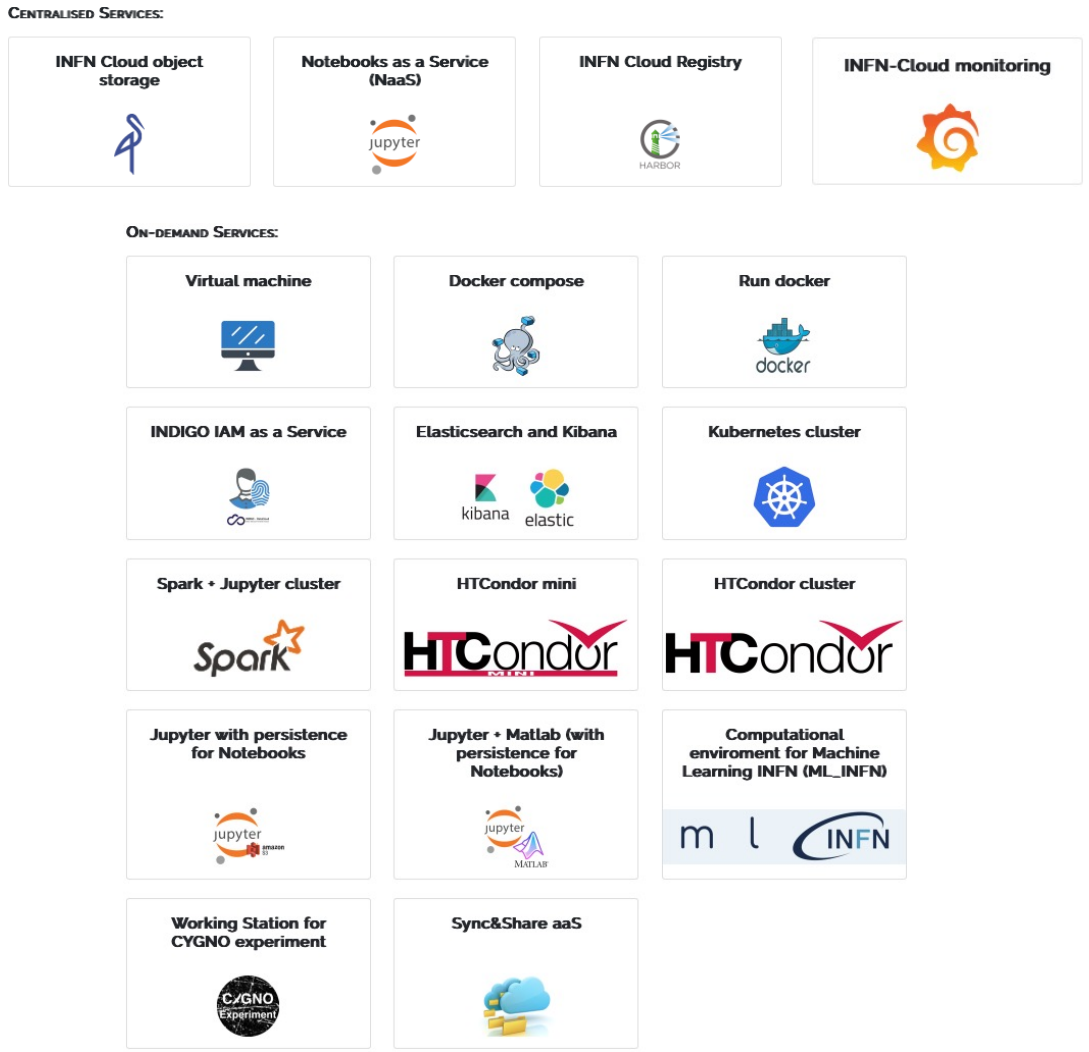
- Start & Stop
- Hostname choice
- Open ports

The INFN Cloud dashboard

https://my.cloud.infn.it

It allows users to:
- access centralized services
- instantiate PaaS services independently





The Infrastructure as Code paradigm

All services are described through an **Infrastructure as Code** paradigm based on a procedural approach, via a combination of:

- **TOSCA** (Topology and Orchestration Specification for Cloud Applications) templates, to model an application stack
- **Ansible** roles, to manage the automated configuration of virtual environments
- **Docker** containers, to encapsulate high-level application software and runtime
- **Helm** charts, to manage the deployment of an application in Kubernetes clusters

It allows to reduce manual processes and increase flexibility and portability across environments

```yaml
node_templates:

 ml_install:
   type: tosca.nodes.DODAS.single-node-jupyterhub
   properties:
     contact_email: { get_input: contact_email }
     iam_url: { get_input: iam_url }
     iam_subject: { get_input: iam_subject }
     iam_groups: { get_input: iam_groups }
     iam_admin_groups: { get_input: iam_admin_groups }
     monitoring: { get_input: enable_monitoring }
     jupyter_hub_image: dodasts/snj-base-jhub:v1.1.1-snj
     jupyter_images: { get_input: jupyter_images }
     jupyterlab_collaborative: { get_input: jupyterlab_collaborative }
     jupyter_post_start_cmd: "/usr/local/share/dodasts/script/post_script.sh"
     jupyterlab_collaborative_image:
       { get_input: jupyterlab_collaborative_image }
     dns_name: { concat: [get_attribute: [HOST, public_address, 0],
     cert_manager_type: { get_input: certificate_type }
   requirements:
     - host: vm_server
```

TOSCA

Ref: [TOSCA Simple Profile in YAML Version 1.1]

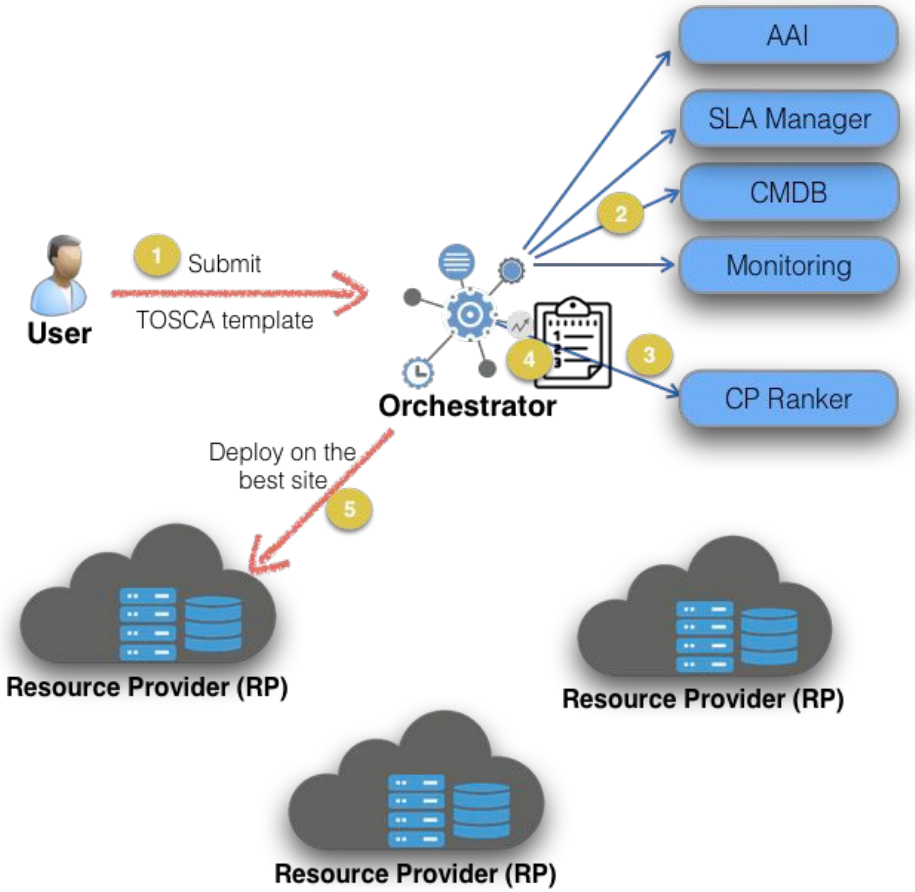
```yaml
artifacts:
  ml_role:
    file: git+https://github.com/DODAS-TS/ansible-role-jupyterhub-env,v2.4.1
    type: tosca.artifacts.AnsibleGalaxy.role
```

```yaml
- name: prepare compose file
  ansible.builtin.template:
    src: jupyter_hub-compose.j2
    dest: /usr/local/share/dodasts/jupyterhub/compose.yaml
  vars:
    iam_client_id: "{{ iam_response.json.client_id }}"
    iam_client_secret: "{{ iam_response.json.client_secret }}"
  when: cert_manager_type != "self-signed"
```

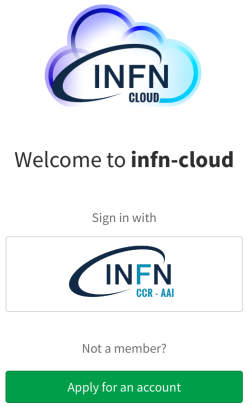
Ansible

```yaml
- name: Run Jupyter Hub
  ansible.builtin.shell:
    cmd: docker-compose up -d
    chdir: /usr/local/share/dodasts/jupyterhub
  when: (run_jupyter | bool)
```

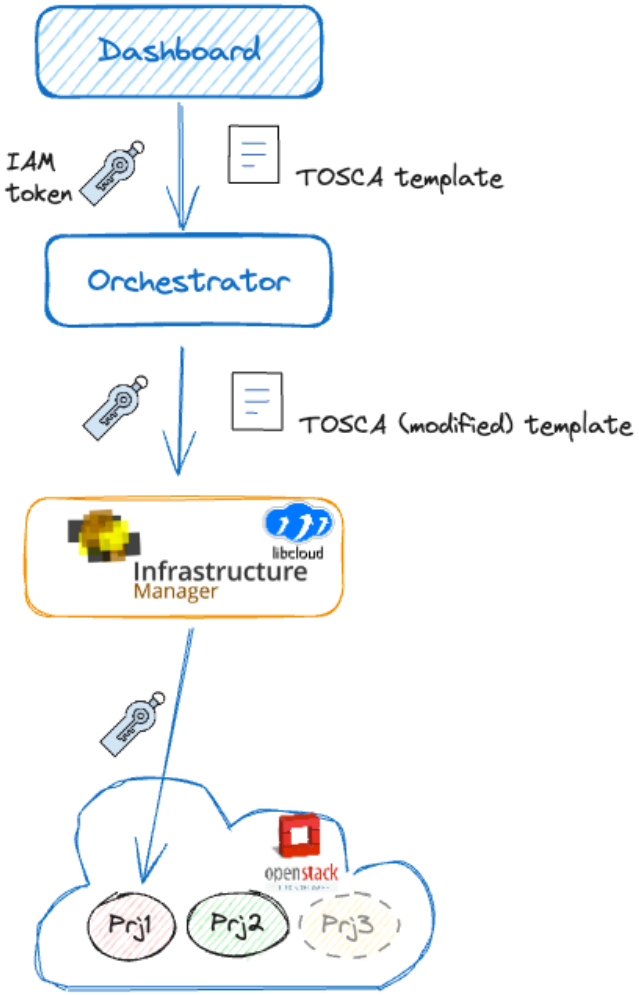
The PaaS Orchestration system



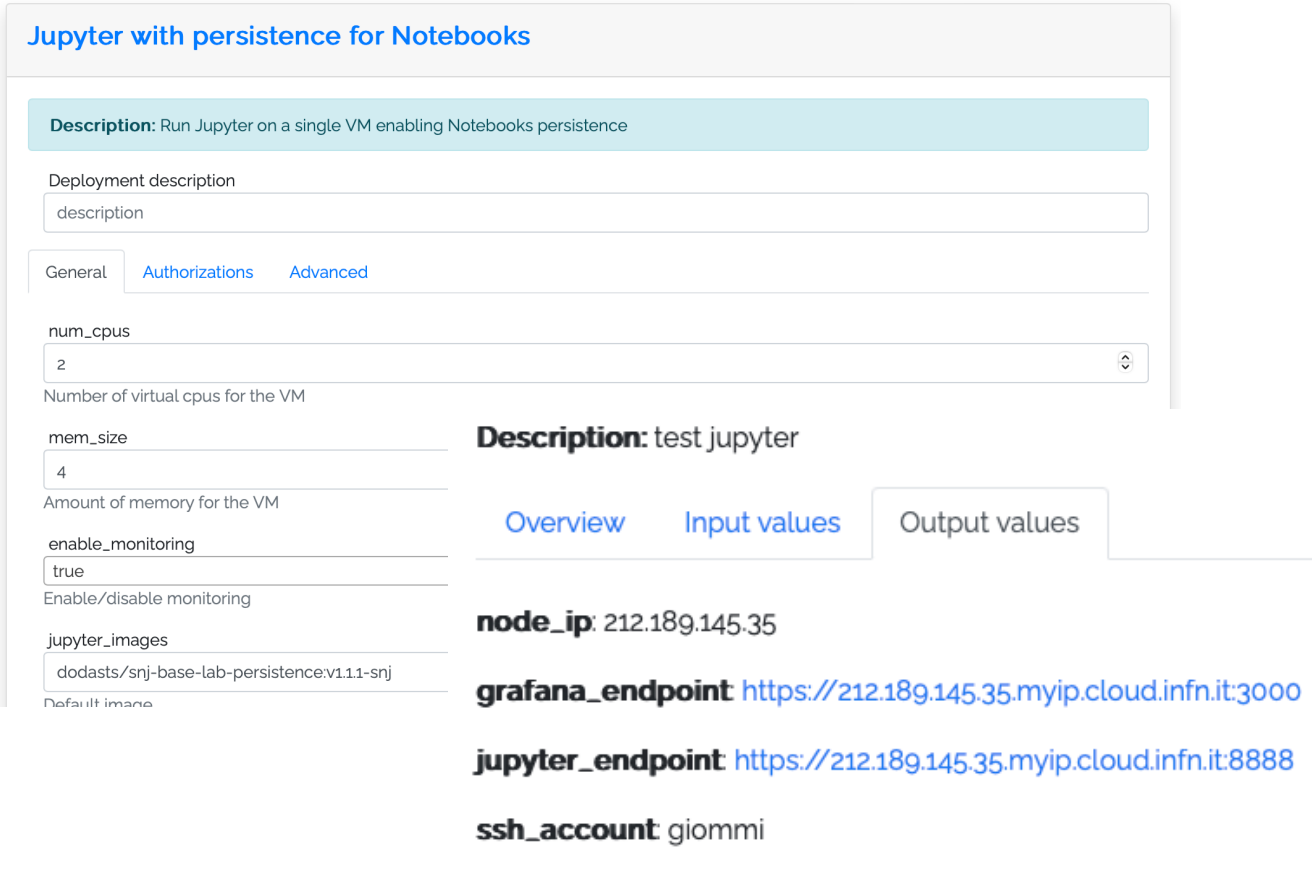
INFN Cloud IAM

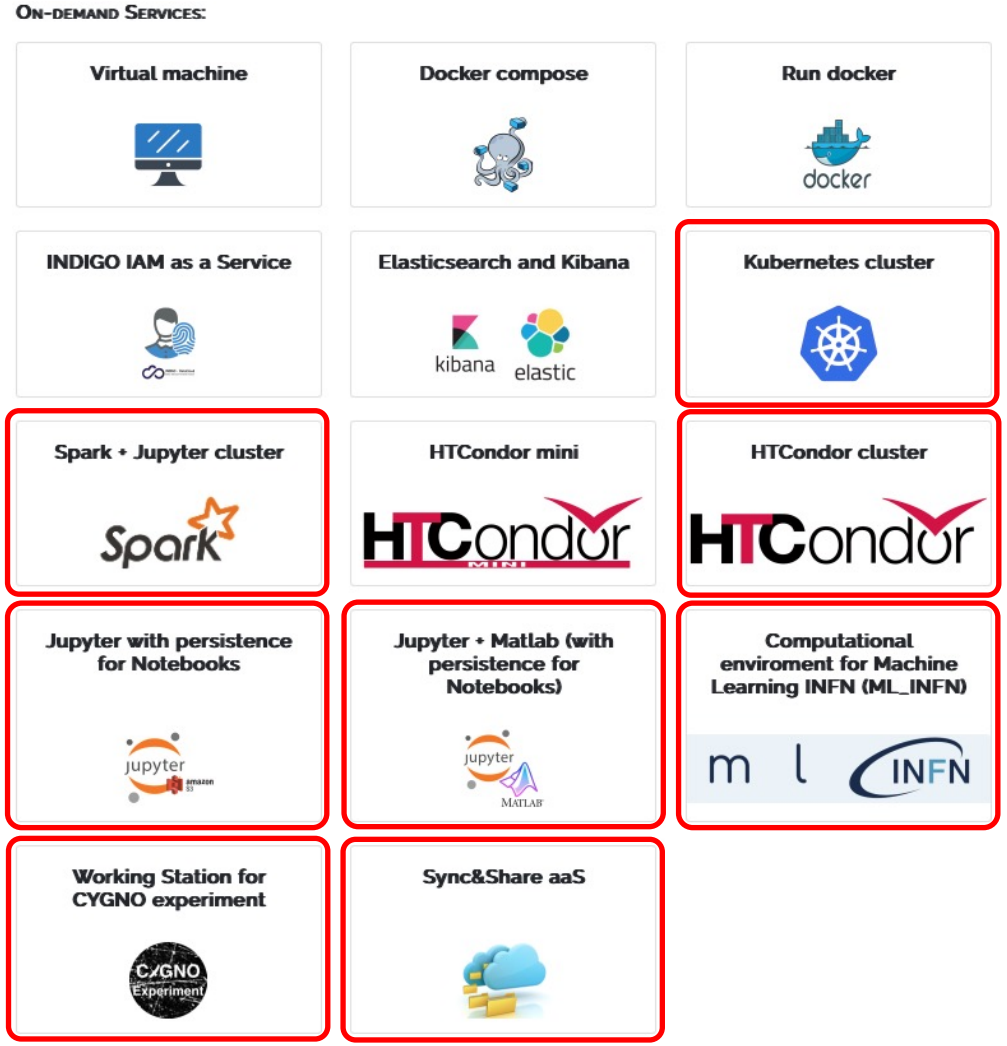


The Orchestrator interacts with the provider services through the **Infrastructure Manager (IM)** for deploying complex and customized virtual infrastructures on IaaS Cloud backends

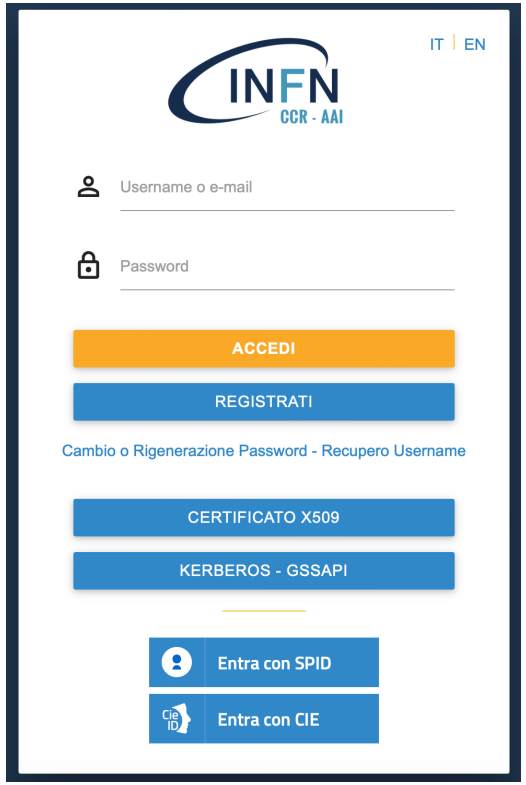


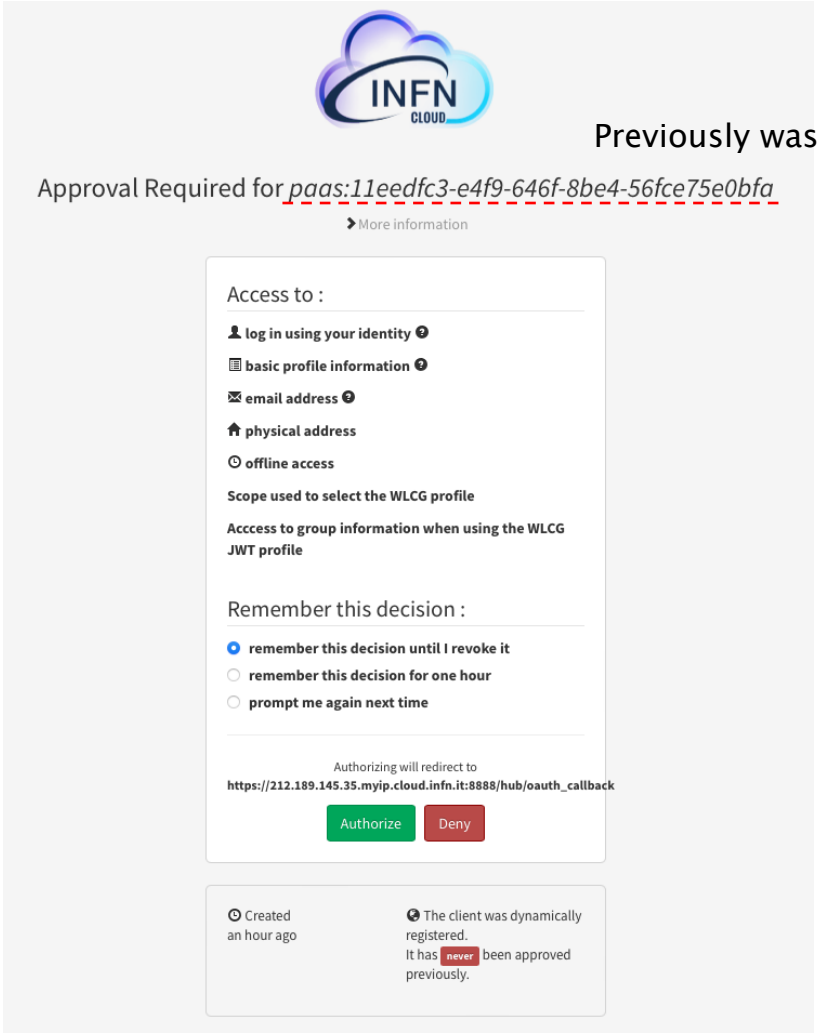
Creation of IAM clients in PaaS services (1)





Creation of IAM clients in PaaS services (2)





Previously was jh-client

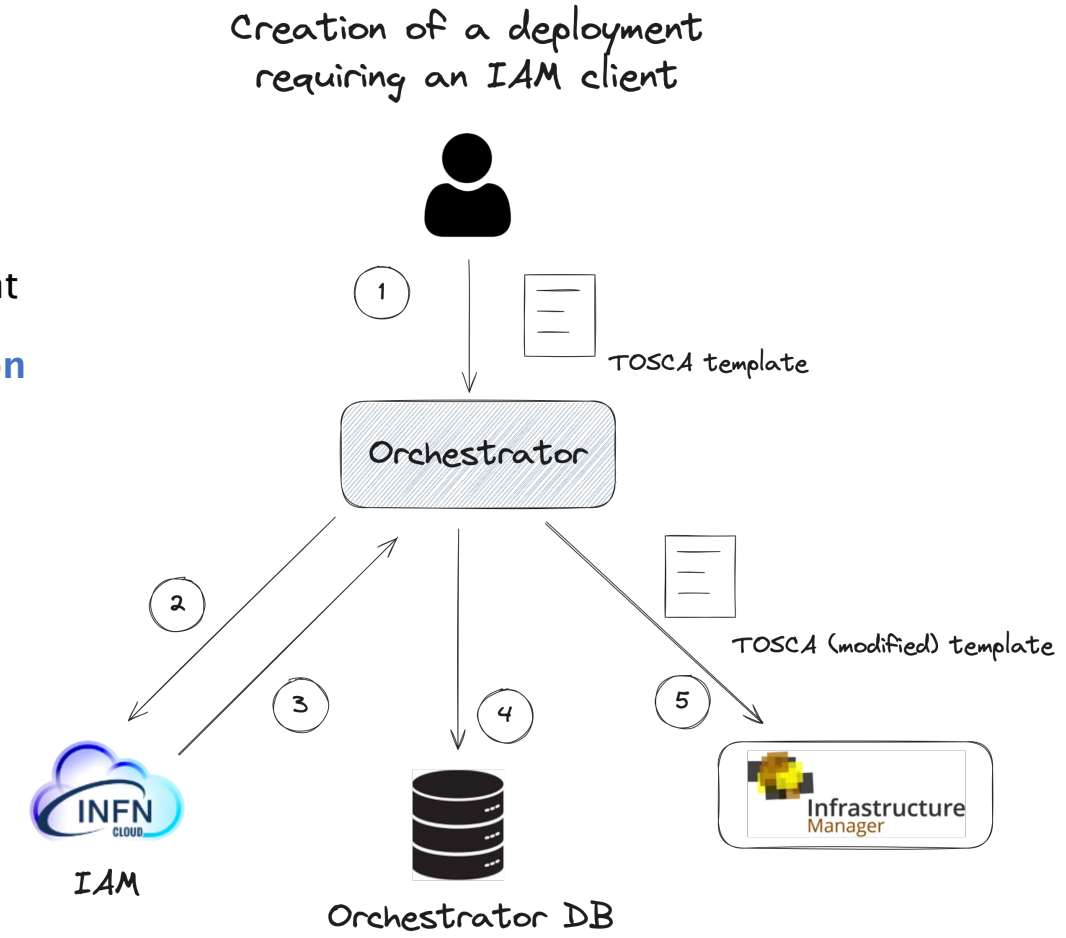
Problem: uncontrolled creation of IAM clients

- An **inventory** of the most frequent IAM client names was made
  - Four «critical» client name, client created by different services/software
- **Most of these clients are no longer used**
  - They are orphaned by the deployments that required their creation as they have been deleted by the user
- Currently the creation of an IAM client is not managed by the PaaS Orchestrator but by the service itself (e.g. in Ansible recipes)
  - When a deployment is deleted, the created IAM client remain
- The increase in the number of IAM clients has led to a **decrease in performance** of the INDIGO IAM service

Client name	Number of created clients
oidc-client	34336
jh-client	597
oc-client	507
ml-client	410

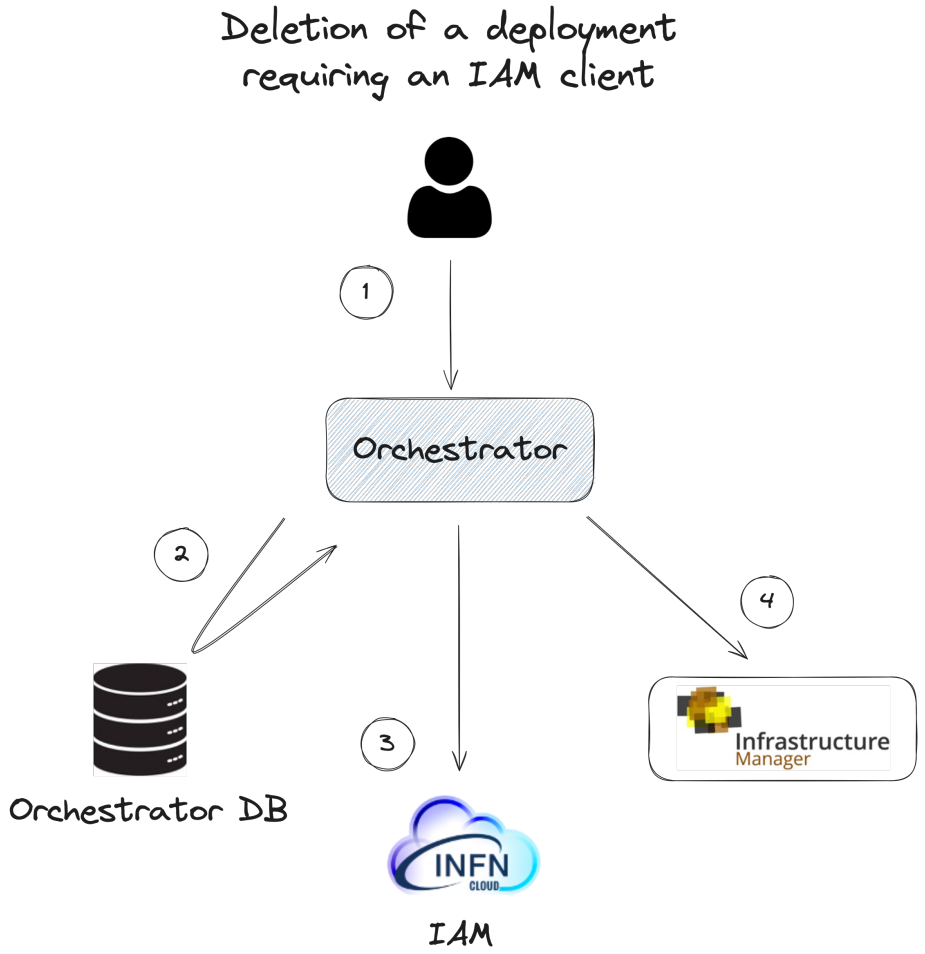
Solution: creation and deletion of IAM clients managed by the PaaS Orchestrator

- Introduced a **TOSCA type** that identifies an IAM client
- Modified the **TOSCA templates** of services that require an IAM client
- Modified the code of the **PaaS Orchestrator** to **manage the creation and deletion of IAM clients**
- Adapted the **Ansible** recipes to the new configuration
- This solution offers users **flexibility**, enabling them to:
  - create multiple clients
  - select the identity provider
  - define scopes
  - assign the client owner



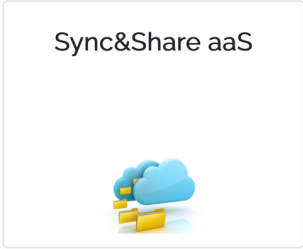
Solution: creation and deletion of IAM clients managed by the PaaS Orchestrator

- Introduced a **TOSCA type** that identifies an IAM client
- Modified the **TOSCA templates** of services that require an IAM client
- Modified the code of the **PaaS Orchestrator** to **manage the creation and deletion of IAM clients**
- Adapted the **Ansible** recipes to the new configuration
- This solution offers users **flexibility**, enabling them to:
  - create multiple clients
  - select the identity provider
  - define scopes
  - assign the client owner

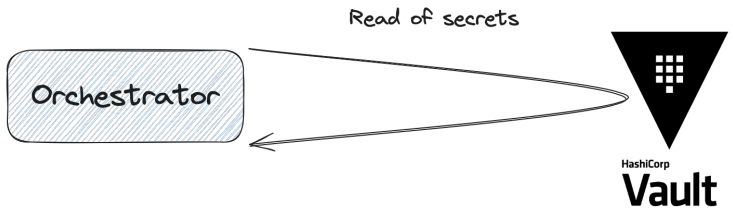


A similar problem: proliferation of S3 buckets

- Some services that integrate **S3 storage** as a backend (e.g. Sync&Share as a Service) need the creation of buckets and usually write a lot of data into them
  - The buckets are created in **Swift** of the INFN Cloud backbone through an **Ansible** recipe
- Currently, when the **deletion of deployments** of these services is triggered, the **S3 buckets remain**
- We implemented a similar solution as for the IAM clients
  - the creation and deletion of the buckets is **managed by the PaaS Orchestrator**
- **But** in this case the aws access key and secret key (necessary for the deletion) cannot be stored in the DB
  - The solution was to use an instance of **HashiCorp Vault** where the Orchestrator can read the secrets




```
name: Create bucket and enable versioning if requested
amazon.aws.s3_bucket:
  aws_access_key: '{{ aws_access_key }}'
  aws_secret_key: '{{ aws_secret_key }}'
  name: '{{ bucket_name }}'
  versioning: "{{ enable_versioning }}"
  state: present
  s3_url: '{{ s3_url }}'
```



Conclusions

- **INFN Cloud**, based on an distributed and federated cloud infrastructure, makes available to its users a set of services through the PaaS Orchestration system
- There are several services that require the creation of **IAM clients** and **S3 buckets**
  - many were orphaned by the deployments that required them and are no longer used
- Our proposed resolution involves **delegating to the PaaS Orchestrator the creation and the deletion of any IAM client and S3 bucket**. This involved:
  - creation of a new TOSCA type
  - update of the TOSCA templates related to the services involved
  - review of the PaaS Orchestrator code to create and delete IAM clients and S3 buckets (and interact with Vault)
  - adaptation of the Ansible recipes of the services to the new configuration

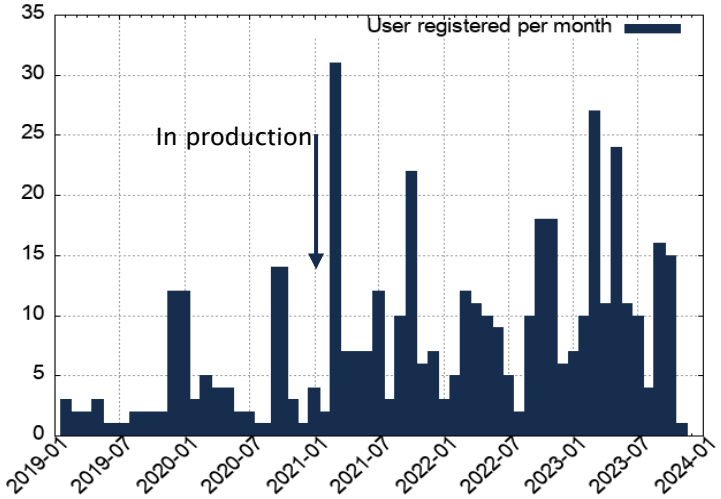
- The presented solution for the management of IAM clients is in production in INFN Cloud
  - The solution for S3 buckets is working but not yet in production

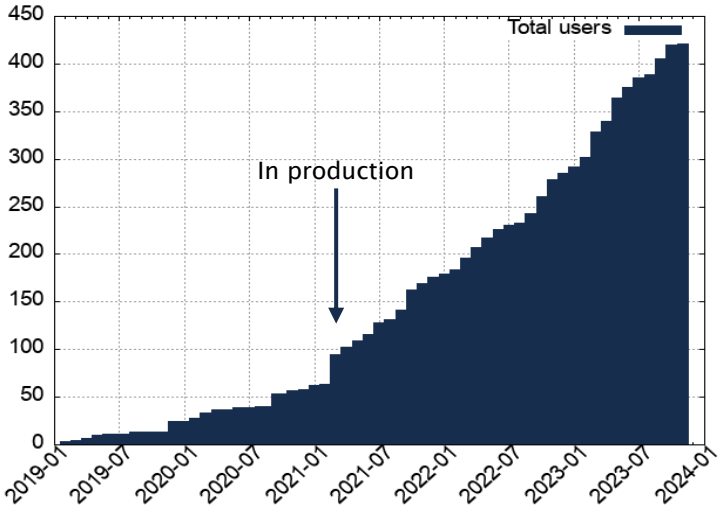
Thank you

luca.giommi@cnaf.infn.it

Users in INFN Cloud

- ML_INFN
- KM3NeT
- ELETBIC
- HERD
- CYGNO
- EUROLABS
- NUCS
- TIFPA
- IXPE
- INCANT
- LHCb
- SI – Sistema Informativo INFN
- MUONE
- QUAX





See Pascolini's talk for more details

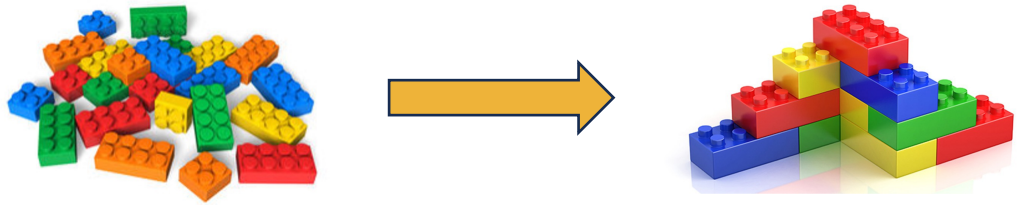
TOSCA: Topology and Orchestration Specification for Cloud Applications

Goals:

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

The INFN Cloud service catalogue is a graphical representation of the **TOSCA templates** that have been developed extending the INDIGO-DC **custom types**

- We are following a lego-like approach, building on top of reusable components and exploiting the TOSCA service composition pattern



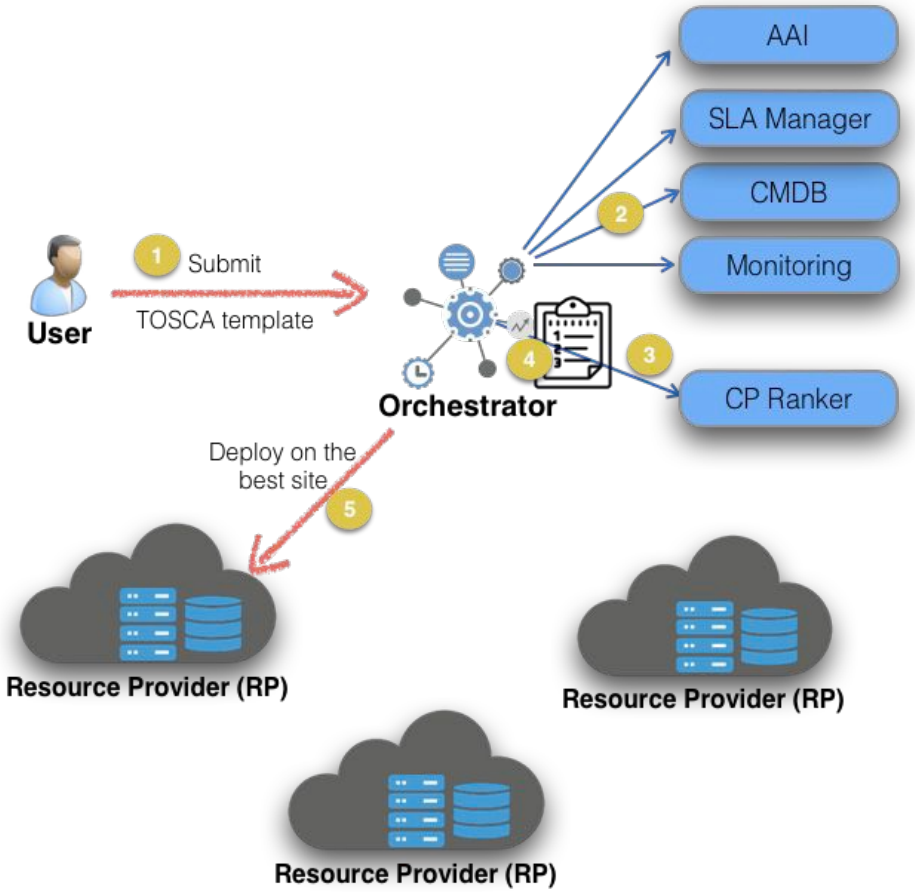
Main objectives:

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

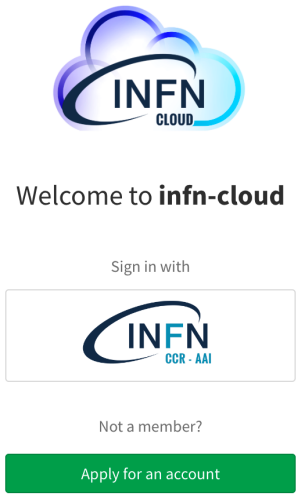
The service catalogue can be easily extended with the simple addition/customization of TOSCA templates.

Ref: TOSCA Simple Profile in YAML Version 1.1

The PaaS Orchestration system



INFN Cloud IAM



Consistent group-based authorization policies are applied at all Cloud levels (IaaS, PaaS, SaaS)

The Orchestrator interacts with the provider services through the **Infrastructure Manager (IM)** for deploying complex and customized virtual infrastructures on IaaS Cloud backends

INDIGO IAM

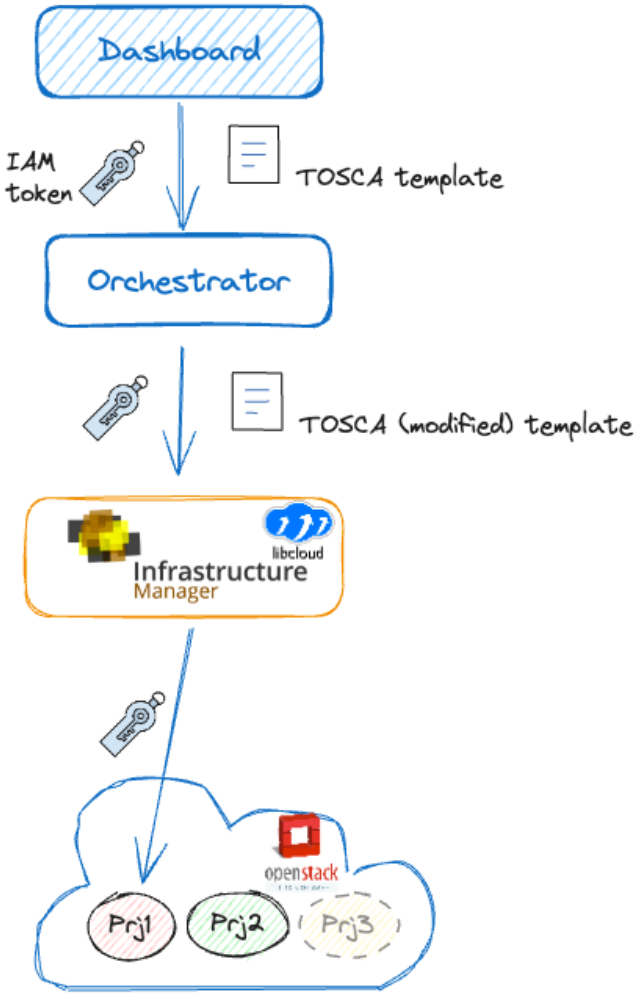
It is the **Identity Access Management** service used in INFN Cloud

We are exploiting the following main capabilities provided by IAM:
- Authorization & Membership: orthogonal to authentication, group-based
- Provide ability for services to act on behalf of users
- Support for long-running applications (token renewal)

Advantages and objectives:
- Allow the federated access to the distributed resources
- Allow to trace the user and link resources to its user
  - important for accounting & in case of security incidents


```
"sub"  "564f8033-4025-4fad-889f-83d01fec157c"
"iss"  "https://iam.cloud.infn.it/"
"name"  "Luca Giommi"
"groups"
  "users"
  "admins/beta-testers"
  "ml-infn"
  "users/naas"
  "users/ml-infn"
  "orchestrator-admin"
  "users/s3"
  "priv-admins"
  "admins"
```

```
"preferred_username"  "giommi"
  "organisation_name"  "infn-cloud"
  "exp"  1710515915,
  "iat"  1710512315,
  "jti"  "2145a6a1-99ad-4a39-8497-d085f3778a7d"
  "client_id"  "69ef2d84-4d8c-4294-83d3-a7c27f77a22d"
  "email"  "luca.giommi@cnaf.infn.it"
```

Solution: creation and deletion of IAM clients managed by the PaaS Orchestrator

- Introduced a **TOSCA type** that identifies an IAM client
- Modified the **TOSCA templates** of services that require an IAM client
- Adapted the Ansible recipes to the new configuration
- Modified the code of **PaaS Orchestrator** to **manage creation and deletion of IAM clients**
  - The Orchestrator receives a template with the request of creation of an IAM client
  - The Orchestrator creates the IAM client (using a reasonable name)
  - The Orchestrator add info about the IAM client to the template and submit it to the IM
  - The Orchestrator saves useful info about the IAM client in its DB
  - When the user triggers the deletion of a deployment, the Orchestrator get back info about the client and deletes the client
- This solution offers users **flexibility**, enabling them to
  - create multiple clients
  - select the identity provider
  - define scopes
  - assign the client owner

```
tosca.nodes.indigo.iam.client:
  derived_from: tosca.nodes.Root
  properties:
    owner:
      descritpion: Id of the user requesting the creation of the client
      required: no
      type: string
    issuer:
      descritpion: Identity provider to be used for the creation of the client
      required: no
      type: string
    scopes:
      description: space delimited strings
      required: no
      type: string
    client_id:
      required: no
      type: string
    registration_access_token:
      required: no
      type: string
```

```
node_templates:

  iam_client:
    type: tosca.nodes.indigo.iam.client
    properties:
      scopes: openid email profile wlcg offline_access address wlcg.groups
      issuer: { get_input: iam_url }
      owner: { get_input: iam_subject }
```

TeRABIT in a nutshell

Create a **distributed**, **hyperconnected**, **hybrid Cloud-HPC** environment that offers services prepared to meet the various research needs.
The environment will be created by **federating**, **integrating** and **updating** the GARR-X, PRACE-Italia and HPC-BD-AI research infrastructures.

Three main objectives:

1. Enable **widespread data transfer**, **up to Terabits** per second, and services on a national scale, with **particular attention to southern and island regions, connected to Europe**
2. **Innovate the central HPC node of PRACE-Italia** while maintaining the Tier-1 level.
3. **Innovate the set of HPC services** offered to researchers, beyond centralized calculation model

https://www.terabit-project.it/

INFN Istituto Nazionale di Fisica Nucleare

OGS Istituto Nazionale di Oceanografia e di Geofisica Sperimentale

Consortium GARR

CINECA