

Infrastructure, Platform and Software as Services in INFN Cloud

Luca Giommi

INFN-CNAF






luca.giommi@cnafe.infn.it



INFN research and structures



216 activities distributed in 33 structures (labs, groups and divisions)

	Particle Physics	17 experiments
	Astroparticle Physics	45 experiments
	Nuclear Physics	23 experiments
	Theoretical Physics	35 initiatives
	Technological Research	96 experiments

Facilities at INFN

- 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.
- All the INFN centers are connected through 10–100 Gbit/s dedicated links via the GARR network.
- 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**
 - <https://www.cloud.infn.it>

Cloud@CNAF
Cloud@ReCaS-Bari
CloudVeneto
Cloud@Torino ...



Work packages and leaders

INFN Cloud is internally organized into 7 Work Packages (WP), run by people belonging to several INFN sites in a fully distributed way

- **WP1** – Operations: *Stefano Stalio (LNGS), Diego Michelotto (CNAF)*
 - Operations management of the backbone infrastructure, monitoring and accounting
- **WP2** – Documentation, User Support, Communication and Training: *Carmelo Pellegrino (CNAF), Emidio Giorgio (LNS)*
- **WP3** – Resources, Data Lake and Sustainability: *Giacinto Donvito (Bari), Daniele Cesini (CNAF)*
- **WP4** – Security and Policies: *Vincenzo Ciaschini (CNAF), Luca Carbone (MIB)*
- **WP5** – Middleware and New Services: *Marica Antonacci (Bari), Enrico Vianello (CNAF)*
- **WP6** – Research and Development, Testbeds, Use Cases: *Daniele Spiga (Perugia), Massimo Sgaravatto (Padova)*
- **WP7** – Integrated Systems Management and Legal Compliance: *Barbara Martelli (CNAF), Nadina Foggetti (Bari)*

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

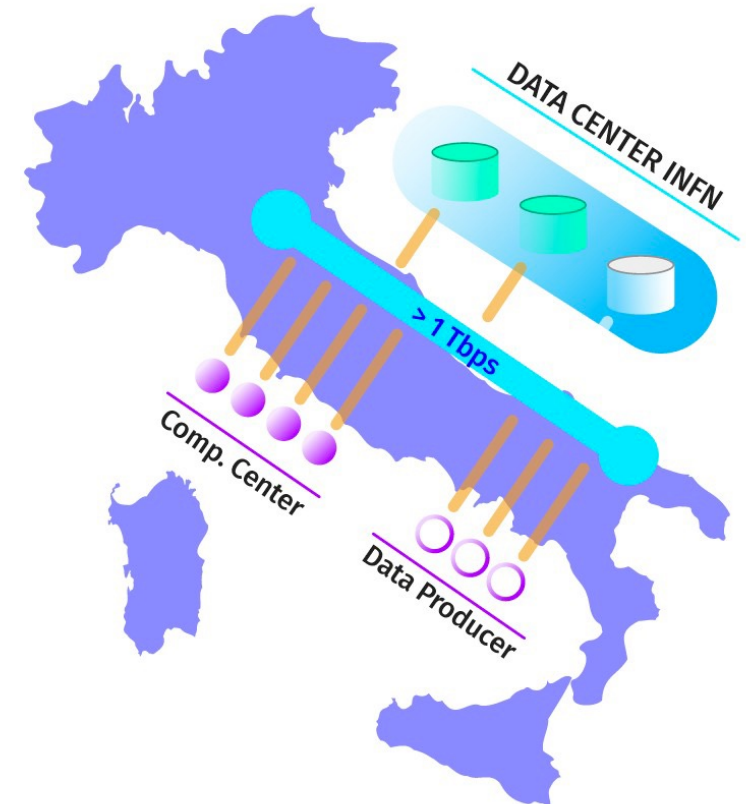
- Backbone's sites are high speed connected and host the INFN Cloud core services
- **Federated clouds**: Cloud@CNAF, CloudVeneto, Cloud@ReCaS-Bari. Coming soon: Catania Napoli, LNGS, Milano, HTC in Tier-2s, HPC bubbles

Backbone

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

Federated Clouds

> 750 vCPU
> 1.7 TB RAM



Portfolio of Services

- NaaS
- Harbor
- Minio

SaaS



- VM
- Docker Compose
- Run Docker
- Elasticsearch&Kibana
- Kubernetes
- Spark
- Jupyter with persistence
- Sync&Share

PaaS



- Start&Stop
- Hostname choice

IaaS



The Infrastructure as Code paradigm

All services are described through an **Infrastructure as Code** paradigm based on a declarative approach, via a combination of **TOSCA** templates (to model an application stack), **Ansible** roles (to manage the automated configuration of virtual environments), and **Docker** containers (to encapsulate high-level application software and runtime). This allows to reduce manual processes and increase flexibility and portability across environments.

```
node_templates:

m1_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], ".myip.cloud.infn.it"] }
    cert_manager_type: { get_input: certificate_type }
  requirements:
    - host: vm_server

pub_network:
  type: tosca.nodes.network.Network
  properties:
    network_type: public

server_pub_port:
  type: tosca.nodes.network.Port
  properties:
    order: 1
  requirements:
    - binding: vm_server
    - link: pub_network

priv_network:
  type: tosca.nodes.network.Network
  properties:
    network_type: private
```

TOSCA

```
server_priv_port:
  type: tosca.nodes.network.Port
  properties:
    order: 0
  requirements:
    - binding: vm_server
    - link: priv_network

vm_server:
  type: tosca.nodes.indigo.Compute
  properties:
    os_users: { get_input: users }
  capabilities:
    endpoint:
      properties:
        ports: { get_input: ports }
  scalable:
    properties:
      count: 1
  host:
    properties:
      num_cpus: { get_input: num_cpus }
      mem_size: { get_input: mem_size }
  os:
    properties:
      distribution: { get_input: os_distribution }
      version: { get_input: os_version }
```

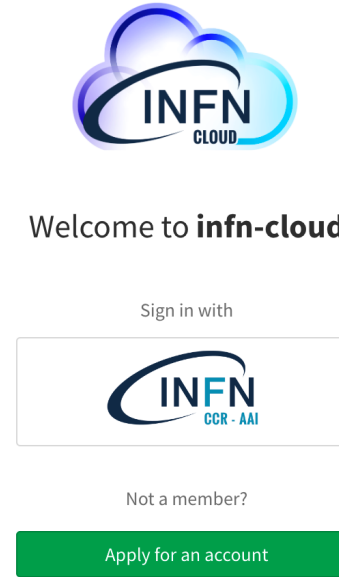
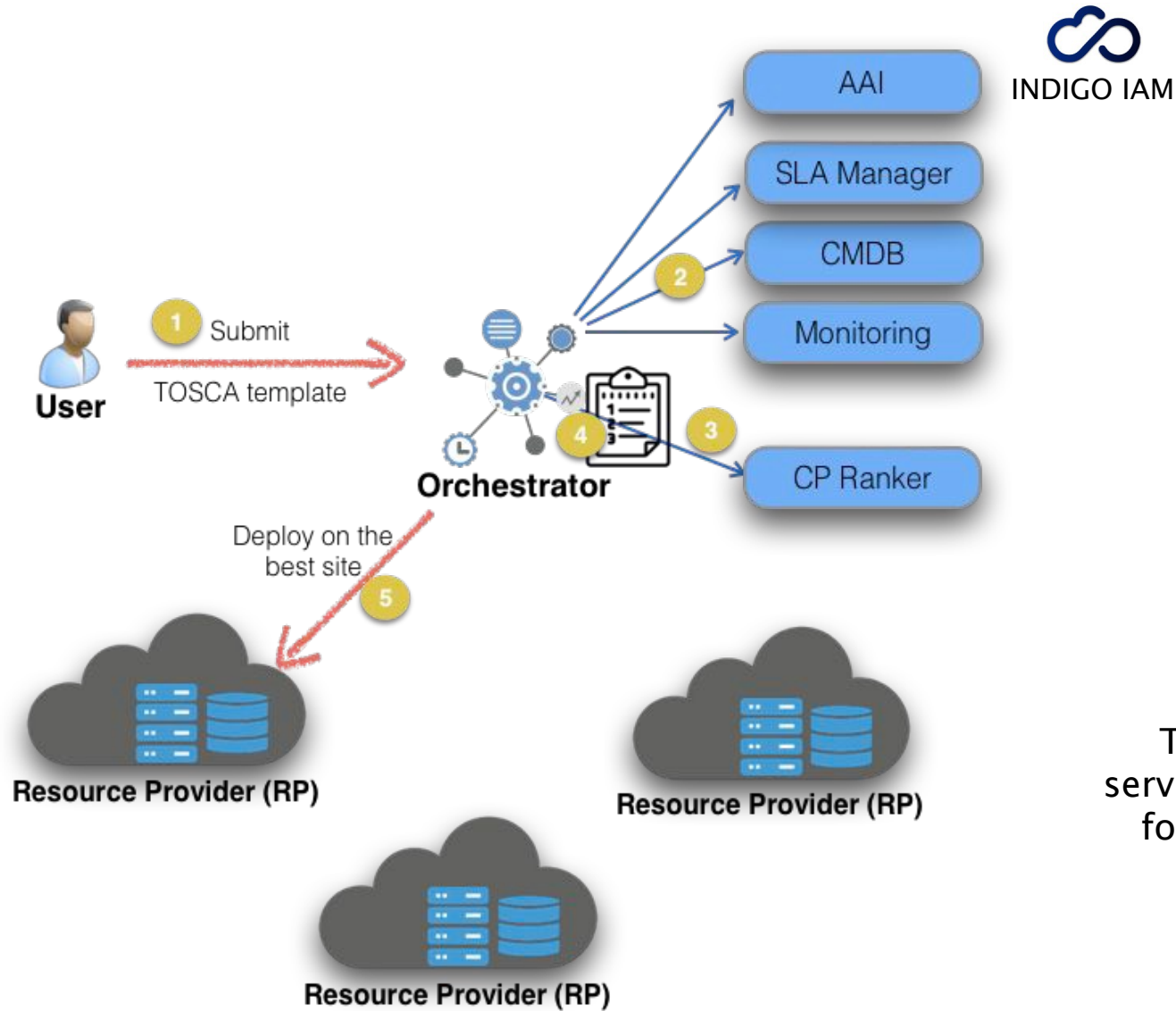
```
- name: Run Jupyter Hub
  ansible.builtin.shell:
    cmd: docker-compose up -d
    chdir: /usr/local/share/dodasts/jupyterhub
    when: (run_jupyter | bool)
```

Ansible

```
- name: register iam client
  uri:
    url: "{{ registration_endpoint }}"
    validate_certs: "no"
    method: POST
    status_code: 201
    headers:
      Content-Type: "application/json"
    body:
      redirect_uris:
        - "https://{{ dns_name }}:{{ jupyter_port }}/hub/oauth_callback"
      client_name: "jh-client"
      token_endpoint_auth_method: client_secret_basic
      scope: openid email profile wlcg offline_access address wlcg.groups
      grant_types:
        - refresh_token
        - authorization_code
      response_types:
        - code
      body_format: json
      return_content: yes
      register: iam_response

- name: Save client info
  copy:
    content: "{{ iam_response.json }}"
    dest: "/usr/local/share/dodasts/jupyterhub/cookies/.client-iam.json"
```


The PaaS Orchestration system



Authorization for services done via groups and subgroups

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

The INFN Cloud dashboard




<https://my.cloud.infn.it>

It allows users to














- access centralized services
- instantiate PaaS services independently



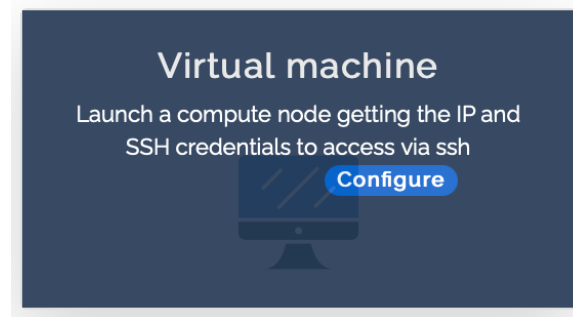
CENTRALISED SERVICES:

INFN Cloud object storage 	Notebooks as a Service (NaaS) 	INFN Cloud Registry 
--	--	--

ON-DEMAND SERVICES:

Virtual machine 	Docker-compose 	Run docker 
INDIGO IAM as a Service 	Elasticsearch and Kibana 	Kubernetes cluster 
Spark + Jupyter cluster 	HTCondor mini 	HTCondor cluster 
Jupyter with persistence for Notebooks 	Computational environment for Machine Learning INFN (ML_INFNO) 	Working Station for CYGNO experiment 
Sync&Share aaS 		

Deploying a virtual machine via dashboard



Deploying a virtual machine via dashboard

Virtual machine

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

[Configure](#)

Select

- VM with no additional storage
- VM with block storage

Attach a volume to the machine

[Submit](#) [Cancel](#)

Deploying a virtual machine via dashboard

Virtual machine

Virtual machine

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

Deployment description

Configuration [Advanced](#)

hostname

service_ports

Ports to open on the host

flavor

Number of vCPUs and memory size of the Virtual Machine

operating_system

Operating System for the Virtual Machine

Select

VM v

VM v

Attach

Deploying a virtual machine via dashboard

The image shows a multi-layered dashboard for deploying a virtual machine. The top layer is a dark blue header with the text "Virtual machine". Below it is a light grey panel with the title "Virtual machine" and a teal description box: "Description: Launch a compute node getting the IP and SSH credentials to access via ssh". Underneath is a "Deployment description" section. A modal window is open over this, titled "Configure scheduling:", with radio buttons for "Auto" and "Manual" (selected). Below that is a "Select a provider:" section with a dropdown menu. The dropdown is open, showing a list of providers: "BACKBONE-BARI: org.openstack.nova" (highlighted in blue), "BACKBONE-CNAF: org.openstack.nova", and "CLOUD-CNAF-T1: org.openstack.nova" (checked with a blue square). Below the provider selection are two dropdown menus: "flavor" with "--Select--" and "operating_system" with "--Select--". At the bottom of the modal are "Submit" and "Cancel" buttons.

List all deployments via dashboard

INFN Cloud Dashboard Deployments ▾ Advanced ▾ External Links ▾ infn-cloud-catchall ▾ Luca Giommi ▾

My deployments

Refresh + New deployment

Show 10 entries Search:

Description ↑↓	Deployment identifier ↑↓	Status ↑↓	Creation time ↑↓	Deployed at ↑↓	Actions ↑↓
1	11edfec8-eda5-be23-gcc9-0242a79ac9f5	CREATE_COMPLETE	2023-05-30 09:04:00	BACKBONE-BARI	Details
prova	11edd3c7-c2b8-3caa-8080-0242a79ac9f5	CREATE_FAILED	2023-04-05 15:37:00	BACKBONE-CNAF	Edit Show template Log Request Ports Manage VMs Lock Delete
mlaas4hep_3	11edb909-d12e-c1de-8080-0242a79ac9f5	CREATE_COMPLETE	2023-03-02 14:52:00	CLOUD-CNAF	
MLaaS4HEP	11edb8dd-2e5e-2b7e-8080-0242a79ac9f5	CREATE_COMPLETE	2023-03-02 09:32:00	CLOUD-CNAF	

Showing 1 to 4 of 4 entries

Previous 1 Next

Orchent: the Orchestrator CLI

<https://indigo-dc.gitbook.io/orchent/>

```
export ORCHENT_TOKEN=<your access token>
or
export ORCHENT_AGENT_ACCOUNT=<your oidc-agent account>
export ORCHENT_URL=<orchestrator_url>
```

usage: orchent <command> [<args> ...]

Commands:

```
help [<command>...]
  Show help.
depls
  list all deployments
depshow <uuid>
  show a specific deployment
decreate [<flags>] <template> <parameter>
  create a new deployment
depupdate [<flags>] <uuid> <template> <parameter>
  update the given deployment
deptemplate <uuid>
  show the template of the given deployment
deplug <uuid>
  get log for given deployment
depdel <uuid>
  delete a given deployment
```

```
./orchent decreate --keepLastAttempt=false --maxProvidersRetry=1 --
user_group dev/cloud ./tosca-templates/single-vm/single_vm.yaml '{
"num_cpus": 1, "mem_size": "2 GB", "users": [{"os_user_add_to_sudoers":
true, "os_user_name": "lgiommi", "os_user_ssh_public_key": "ssh-rsa
xxx"}] }'
```

```
Deployment [11ee7cc7-92ba-4180-94e6-2aab17434343]:
  status: CREATE_IN_PROGRESS
  creation time: 2023-11-06T17:11+0000
  update time: 2023-11-06T17:11+0000
  outputs:
  {}
```

```
./orchent depls
```

```
retrieving deployment list:
  page: 0/1 [ #Elements: 4, size: 10 ]
  links:
    self [http://localhost:8080/deployments {?createdBy,userGroup}]
```

```
Deployment [11ee7cc7-92ba-4180-94e6-2aab17434343]:
  status: CREATE_IN_PROGRESS
  creation time: 2023-11-06T17:11+0000
  update time: 2023-11-06T17:11+0000
Deployment [11ee5dd6-e883-cb85-9501-7e5000df9a09]:
  status: CREATE_COMPLETE
  creation time: 2023-09-28T08:13+0000
  update time: 2023-09-28T08:22+0000
```

How to access INFN Cloud services

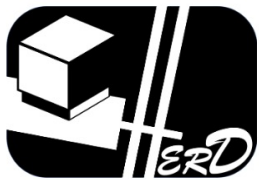
- When the user requests an **account on IAM** a ticket is automatically opened
- User support (WP2) interacts with the user through this ticket
 - verification of requirements
 - approval of the account request
 - addition to groups that do not require the system administrator designation
- If needed the user can request the **system administrator designation**
 - once obtained, forwards the digitally signed PDF to user support for approval and inclusion in groups requiring the designation
- https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/getting_started/getting_started.html#using-infn-cloud

- **Users can** access
 - centralized SaaS services
 - services managed by other system administrator users
- **Users cannot**
 - instantiate or destroy services
 - operate as administrators on services instantiated by others

- **System administrator users can**
 - instantiate and destroy PaaS services hosted on INFN Cloud
 - give other INFN Cloud users access to these services
 - keep deployments updated with the latest security patches

Users in INFN Cloud

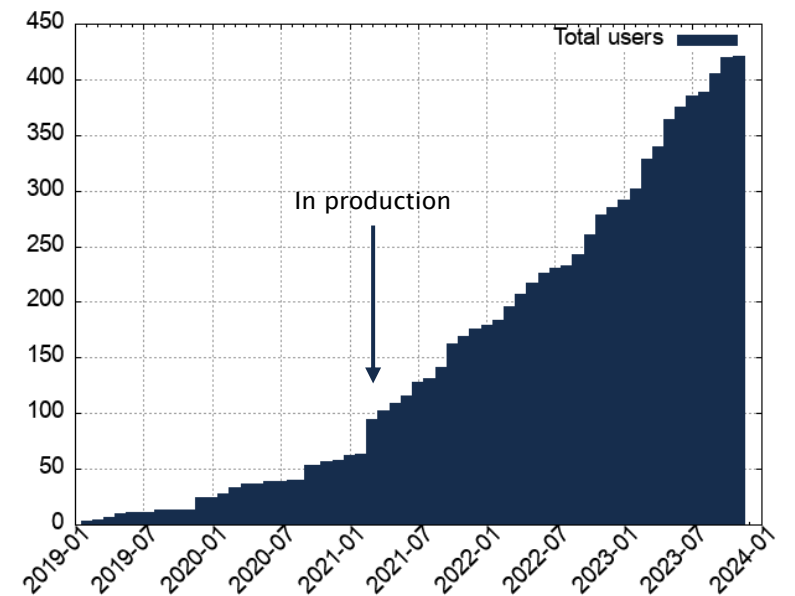
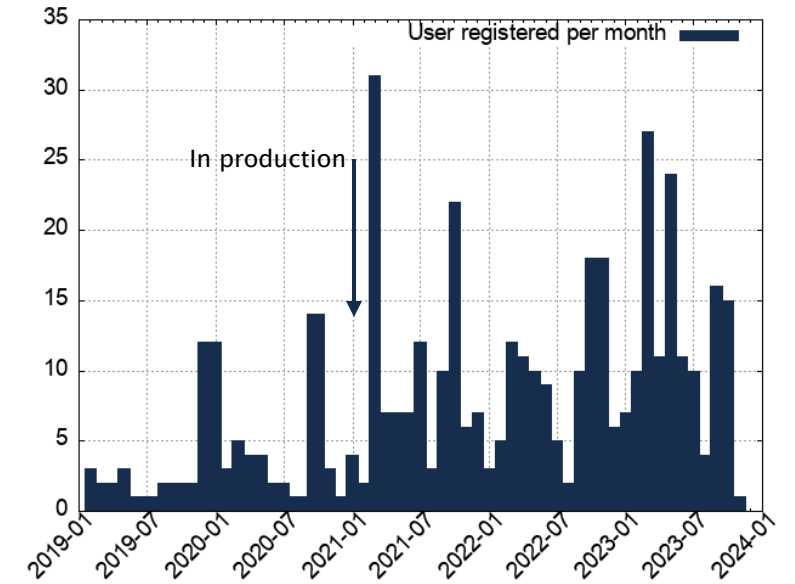
- ML_INFN
- KM3NeT
- ELETBIC
- HERD
- CYGNO
- EUROLABS
- NUCS
- TIFPA
- IXPE
- INCANT
- LHCb
- SI – Sistema Informativo INFN
- MUONE
- QUAX



Trento Institute for Fundamental Physics and Applications



Fifth ML-INFN Hackathon: Advanced Level





INFN Cloud is the architectural foundation for the **evolution of the distributed infrastructure** managed and operated by INFN (HPC-BD-AI).

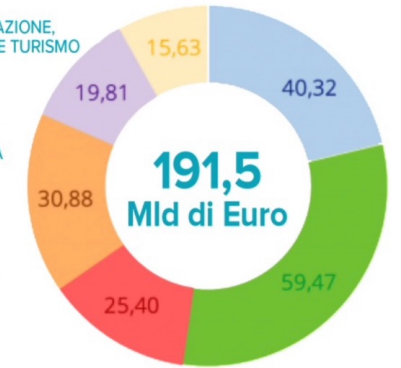
- This is true for all the INFN computing-related engagements with PNRR projects, such as ICSC, **TeRABIT**, DARE, as well as other projects, such as Health Big Data.
- This covers **both hardware resources and the Cloud service portfolio**, in accordance with the service composition architecture. Concretely, this means that we are:
 - expanding hardware resources across the entire INFN DataCloud
 - extending the number of ISO-certified DataCloud regions in Italy
 - increasing the Cloud-native solutions offered by INFN Cloud

PNRR



Piano Nazionale
di Ripresa e Resilienza

Le 6 missioni





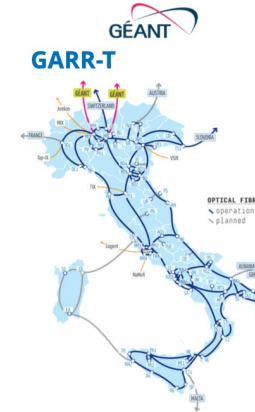
TeRABIT – <https://www.terabit-project.it>

Investment line: Realization of an integrated system of research and innovation infrastructures

Action: Creation of new research infrastructures, strengthening existing ones and their collaboration for scientific excellence in Horizon Europe

- Create a distributed, hyper-connected, hybrid HPC-Cloud environment that offers services designed to meet the needs of research and innovation.
- The environment will be federated in services at various levels and will update the **GARR-T**, **PRACE-Italy** and **HPC-BD-AI** research infrastructures, with the possibility of connections to other national and European research infrastructures and data spaces.

Le infrastrutture di Ricerca partecipanti

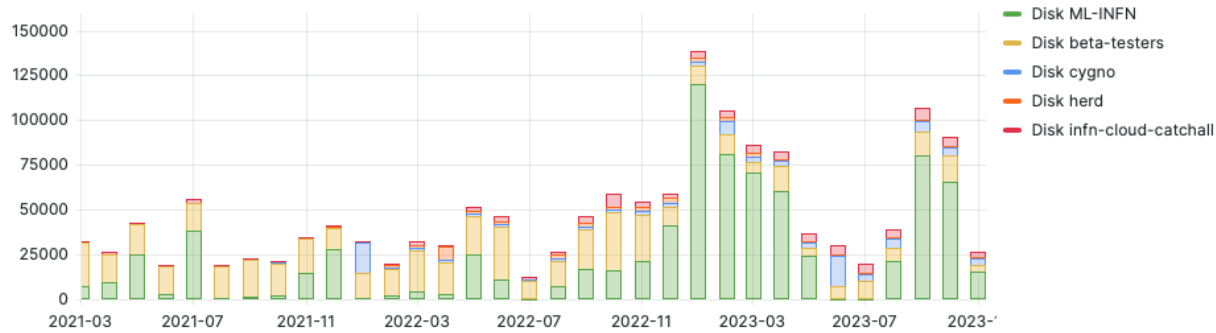


INFN/GARR	GARR	OGS	INFN	OGS
WP 1 Project management	WP 2 Italian Terabit network	WP 3 PRACE Italy	WP 4 Distributed federated cloud	WP 5 Training and dissemination
A1.1 Project Management	A2.1 Acquisition of Optical Fibre and Marine spectrum	A3.1 HPC infrastructure requirements and codesign	A4.1 Deployment of HPC bubble (North)	A5.1 Exploitation and training of TeRABIT integrated infrastructure.
A1.2 Scientific Management	A2.2 Transmission layer and Open Line system	A3.2 HPC infrastructure evolution and deployment	A4.2 Deployment of HPC bubble (South)	A5.2 Dissemination of TeRABIT integrated infrastructure
	A2.3 Packet Network and Network control		A4.3 Implementation of the PaaS orchestration layer	
	A2.4 Control and Services tailoring provision		A4.4 Deployment of flexible cache solutions	

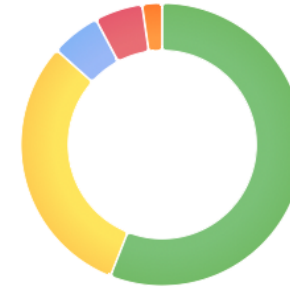
**Thanks for the attention
Questions?**

Resource usage

Ephemeral Disk Used (GB, per month)

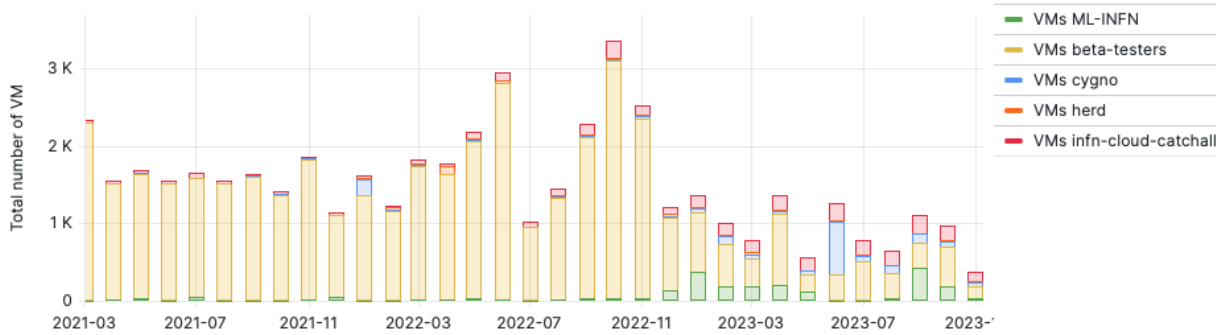


Total Disk Used (GB by project)

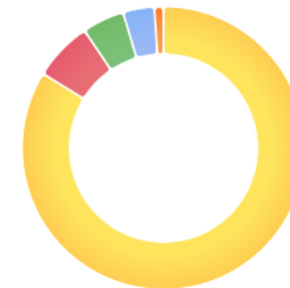


	Value	Percent
Disk ML-INFN	935 K	56%
Disk beta-testers	511 K	31%
Disk cygno	93 K	6%
Disk infn-cloud-catchall	90 K	5%
Disk herd	37 K	2%

Number of VM (instances, per month)



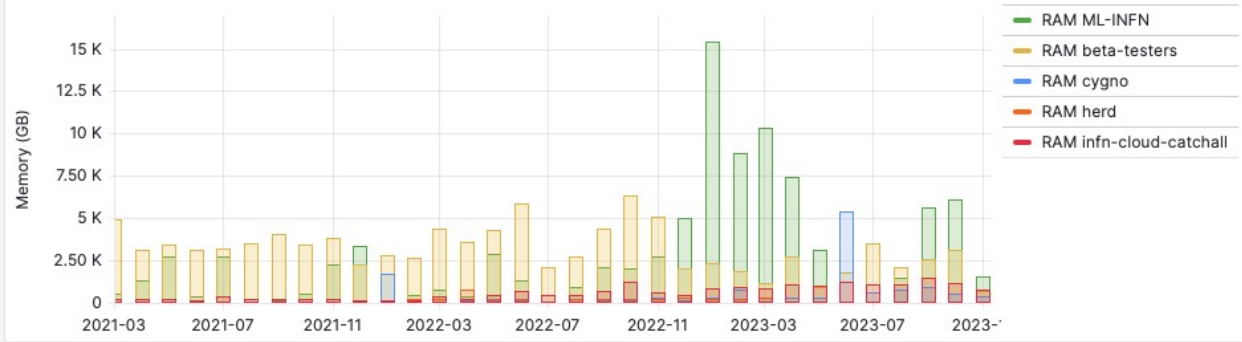
Total number of VM (by project)



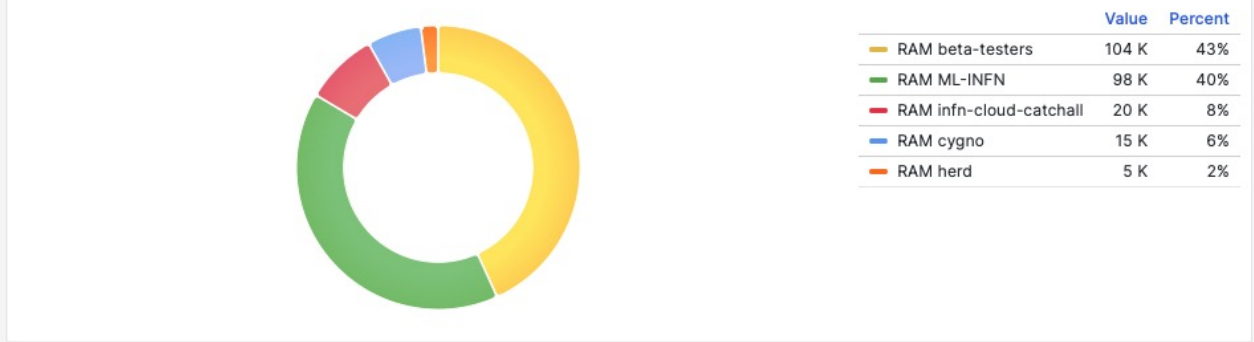
	Value	Percent
VMs beta-testers	42 K	84%
VMs infn-cloud-catchall	3 K	7%
VMs ML-INFN	2 K	5%
VMs cygno	2 K	4%
VMs herd	487	1%

Resource usage

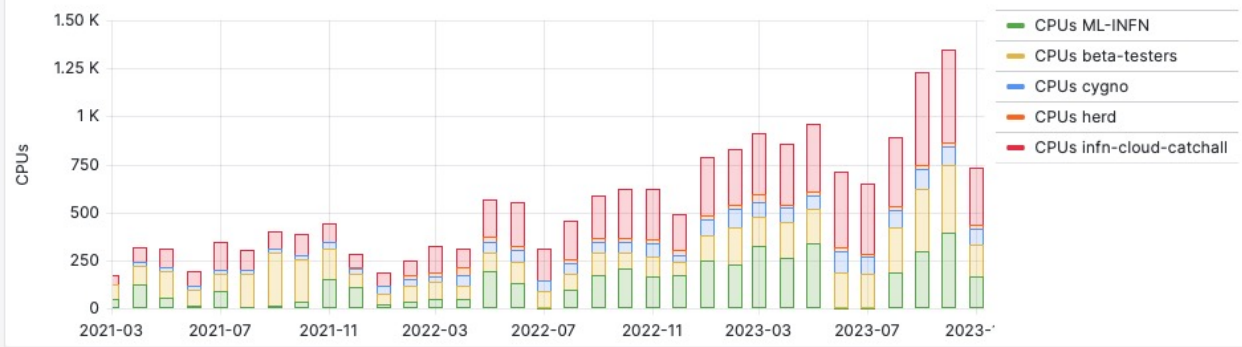
Memory used (GB, per month)



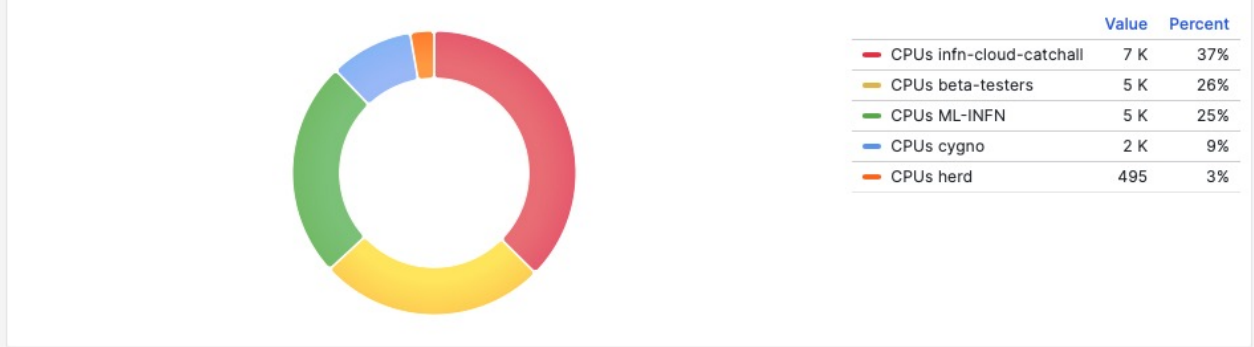
Total Memory used (GB by project)



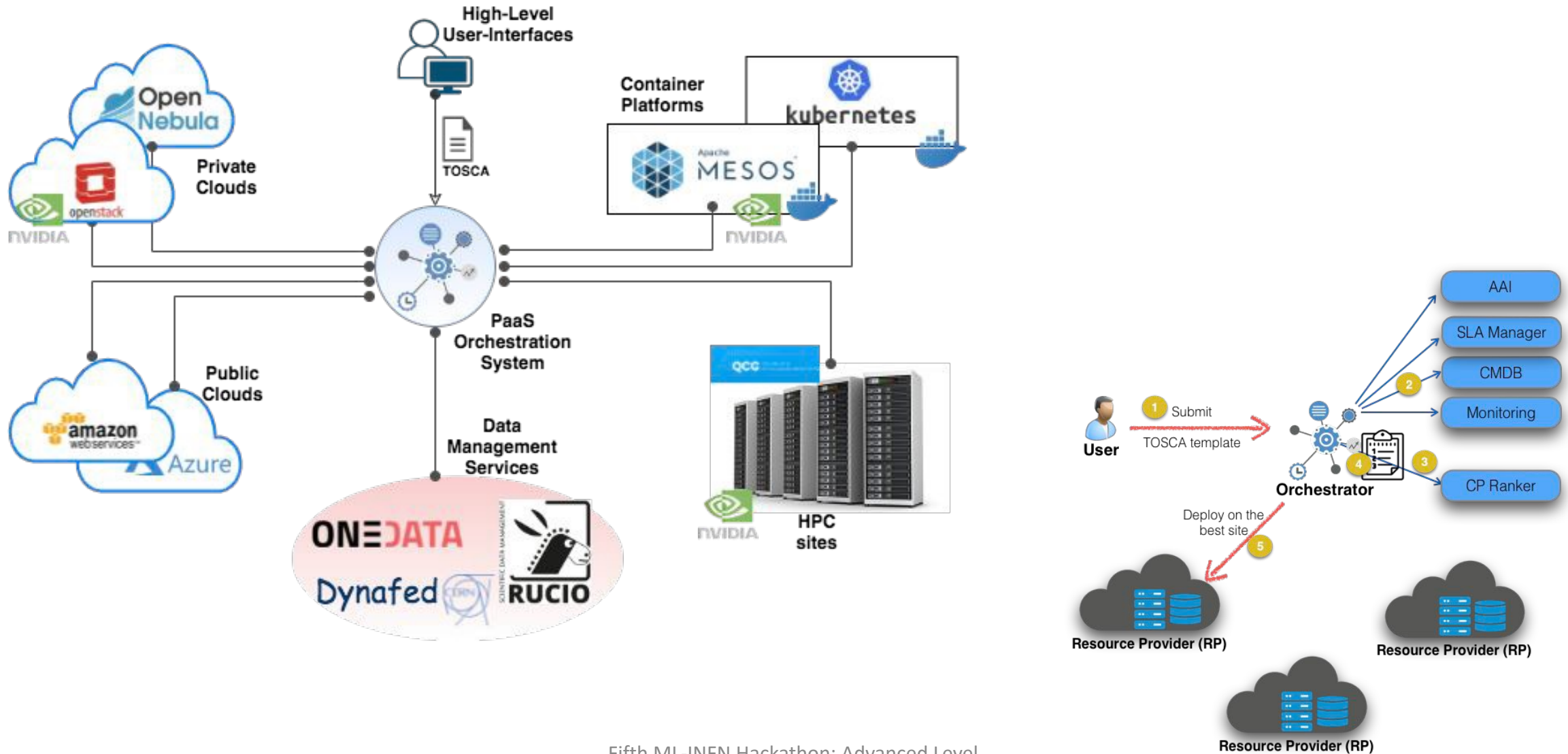
CPU used (CPU, per month)



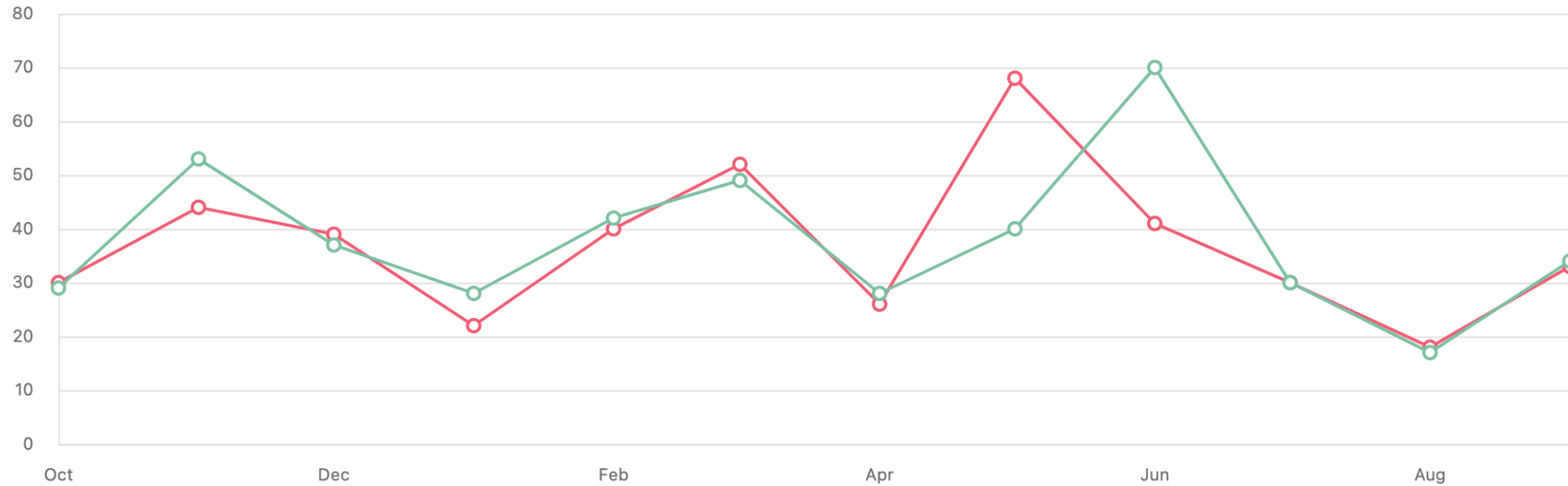
Total CPUs used (by project)



The PaaS Orchestration System



Created vs solved tickets in the last year



Approximately 1260 opened (and managed) tickets in just over 2 years



TeRABIT Work Package 4 – Distributed HPC Cloud

General vision objective

Achieve transparent integration between distributed compute and storage resources covering the Edge, Cloud and traditional HPC domains. This integrated architecture is fundamental to achieve scalable and efficient processing of large amounts of data that flexibly supports different use cases.

Technological objective

Expand the INFN HPC-BD-AI infrastructure to make a series of HPC services available to users on the Cloud (which in TeRABIT they are called HPC Bubbles).

Ongoing activities for the PaaS orchestration layer

- Ai-based orchestrator
- New implementation of the CMDB
- Creation and deletion of IAM clients and S3 buckets managed by orchestrator