How to deploy containers on INFN-CLOUD

Corso base su *Docker* - September 5-7 2022

Marica Antonacci (INFN BA)

What is INFN-Cloud?

INFN Cloud is an internal project which aims to

- manage a (large) fraction of the INFN resources in a sustainable and optimized way;
- make different INFN communities able to access resources, regardless of the availability of local and dedicated hardware (including special hw like GPUs), of the availability of IT skilled people;
- focus on high-level added value services, not on “infrastructures”, to support:
  - ➢ Scientific Computing
  - ➢ Development and R&D, testing of new services
  - ➢ Training activities
  - ➢ Support to INFN data centers (for example for backups of services, etc)

INFN Cloud is built on top of INFN experiences, know-how and solutions developed during several projects and initiatives.

The INFN Cloud architecture

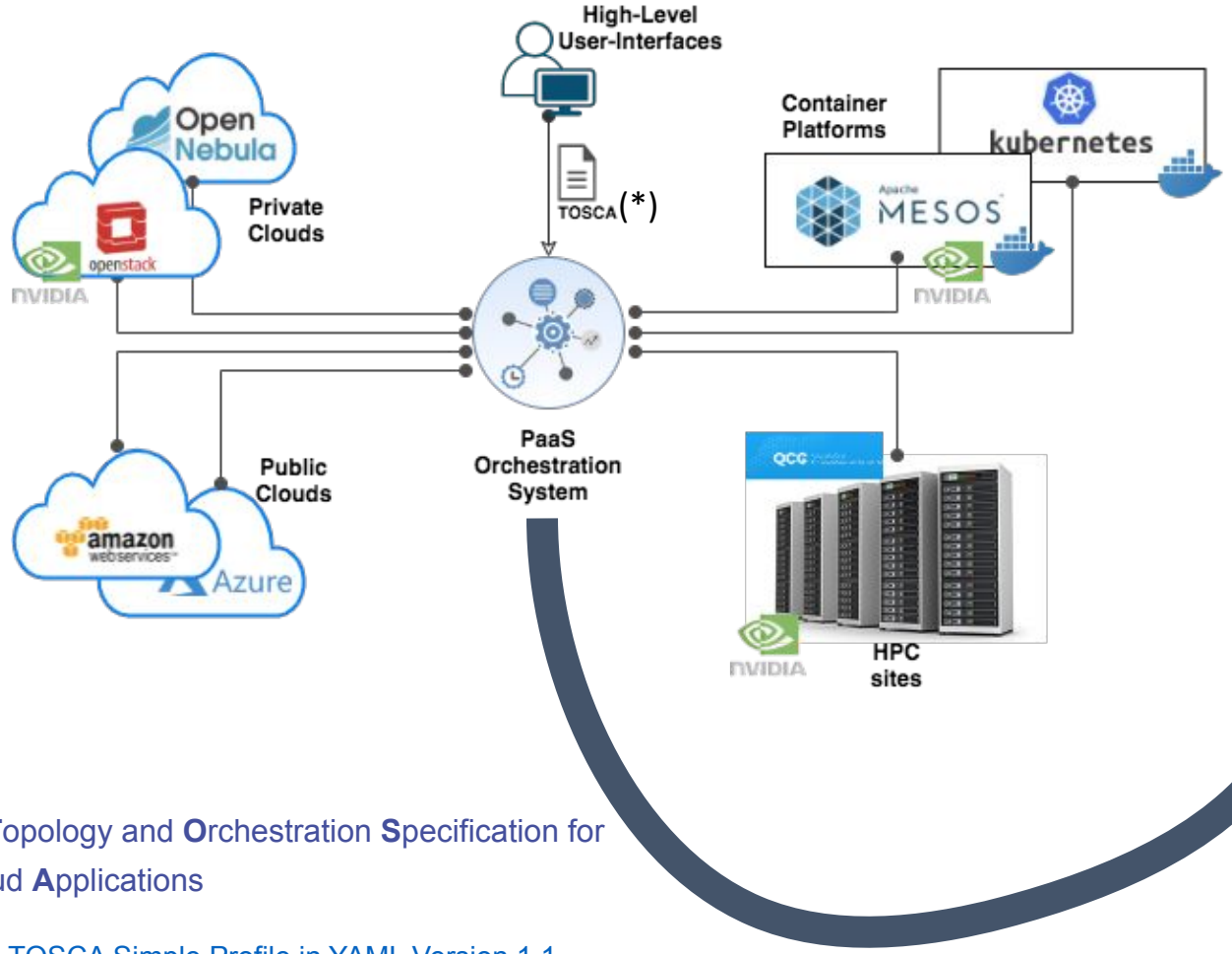
Architecturally INFN Cloud is a **federation** of existing infrastructures

- **the INFN Cloud backbone**, that consists of two tightly coupled federated sites: BARI and CNAF
- **a scalable set of satellite sites**, geographically distributed across Italy, and loosely coupled.
  - Currently Cloud@CNAF, CloudVeneto and ReCaS-Bari are federated with the backbone

Key enabling factors for the federation:

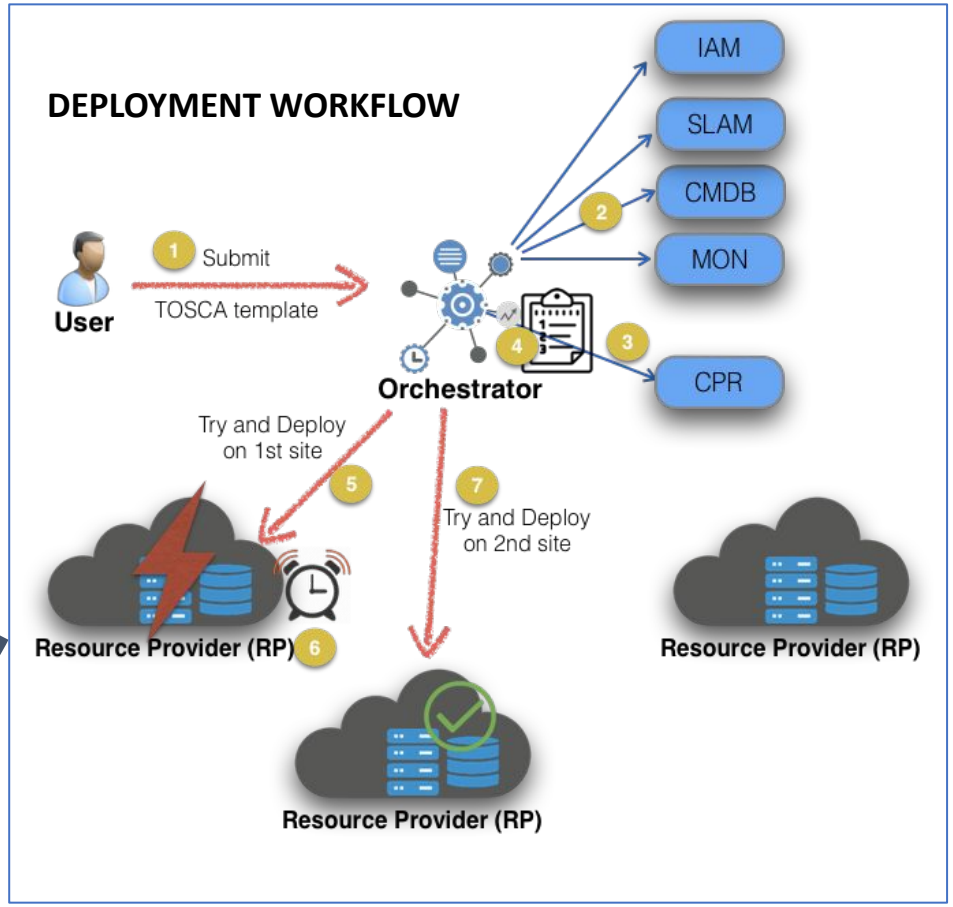
- leverage the same authentication/authorization layer based on **INDIGO-IAM**
- agree on a consistent set of policies and **participation rules** (user management, SLA, security, etc.)
- transparent and dynamic orchestration of the resources across all the federated infrastructures through the **INDIGO 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]



The INFN-Cloud services

Virtual Machines (VM) possibly with external volume for storing data.

Docker containers

Pre-configured environment for **data analytics**

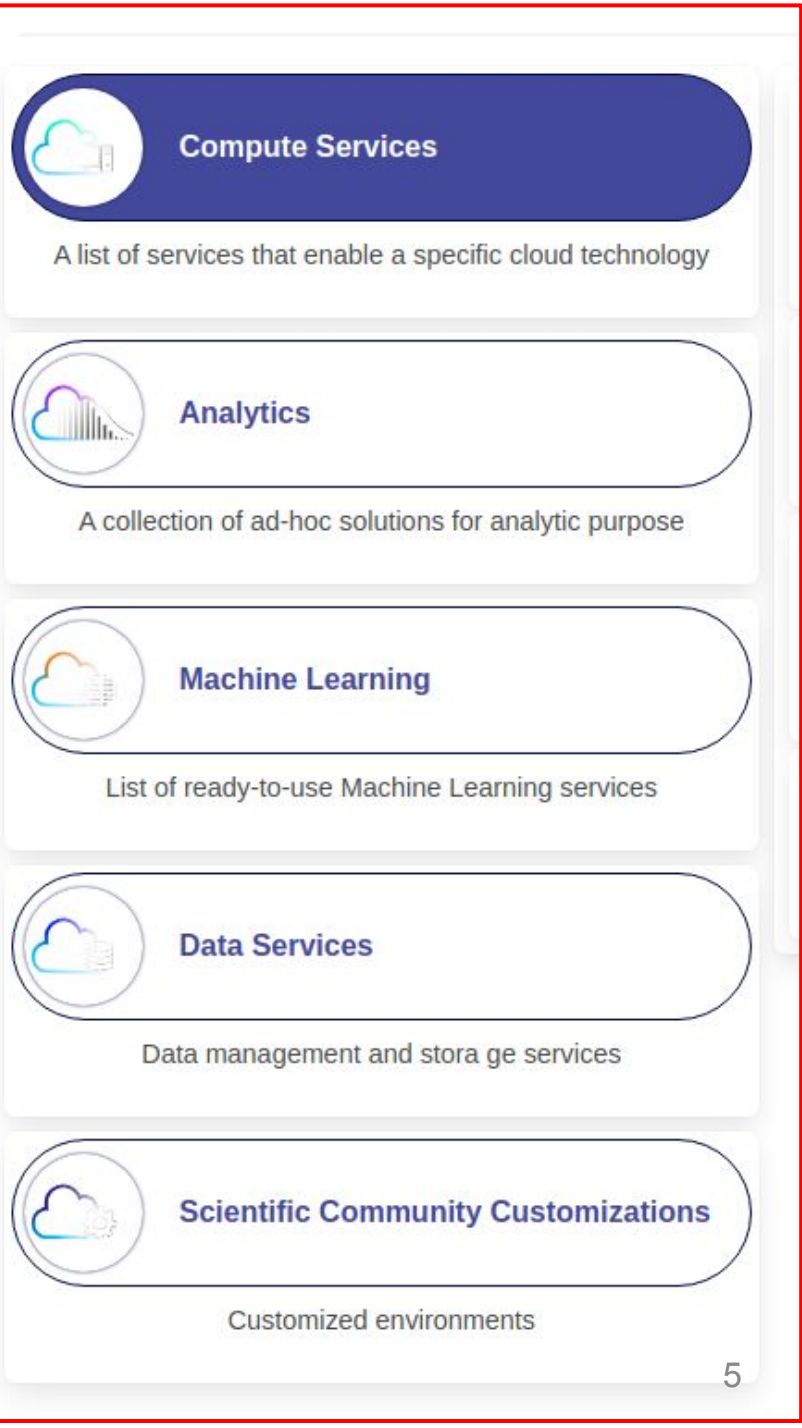
- Spark e/o ElasticSearch e Kibana, R, etc..

Storage solutions: Object storage/posix, possibly connected to high level application layers;

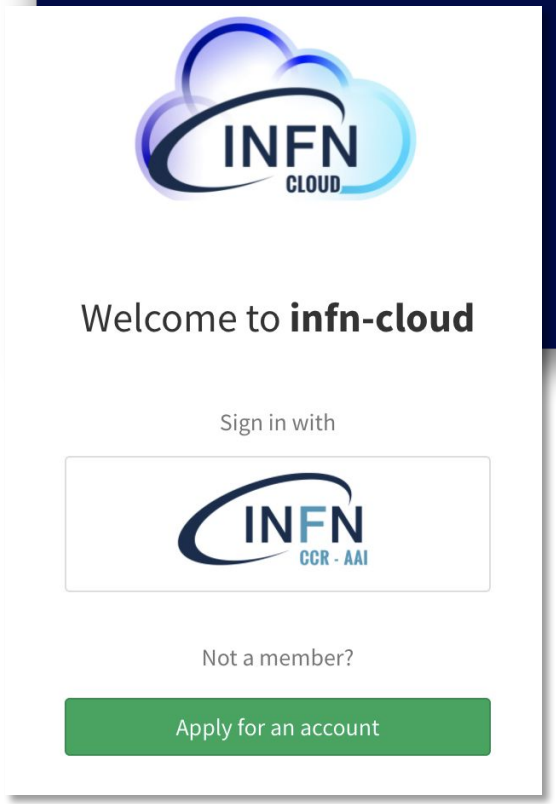
- Jupyter Notebooks with persistent storage (replicated)

Dynamic Clusters even designed and tuned taking into account the specific communities needs;

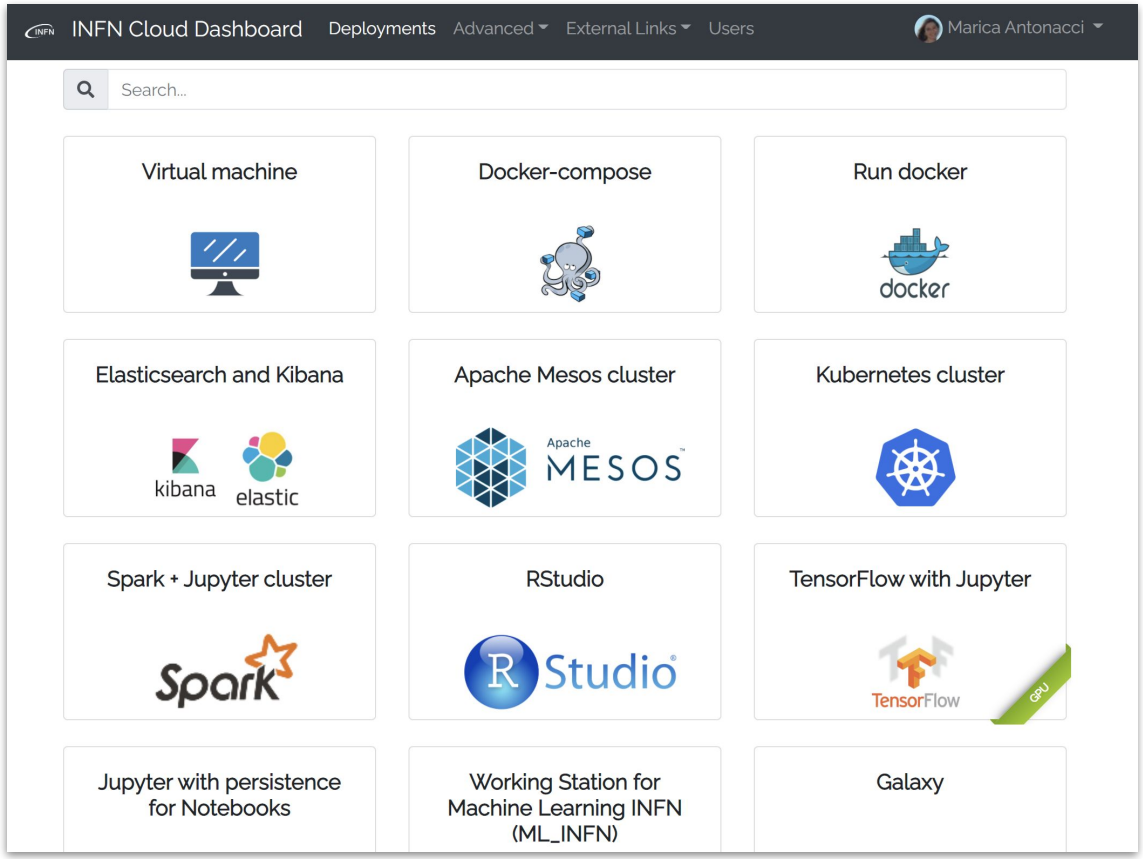
- HTCondor batch system; environment optimized for ML i.e. equipped with GPUs
- Container orchestrators such as K8s and Mesos



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.

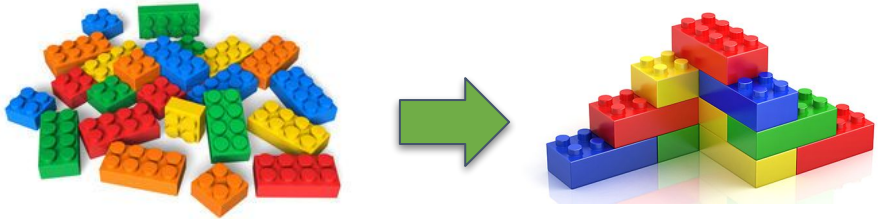
The service catalogue

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



Available services

harbor.cloud.infn.it

minio.cloud.infn.it

hub.cloud.infn.it

Fully-Managed Services

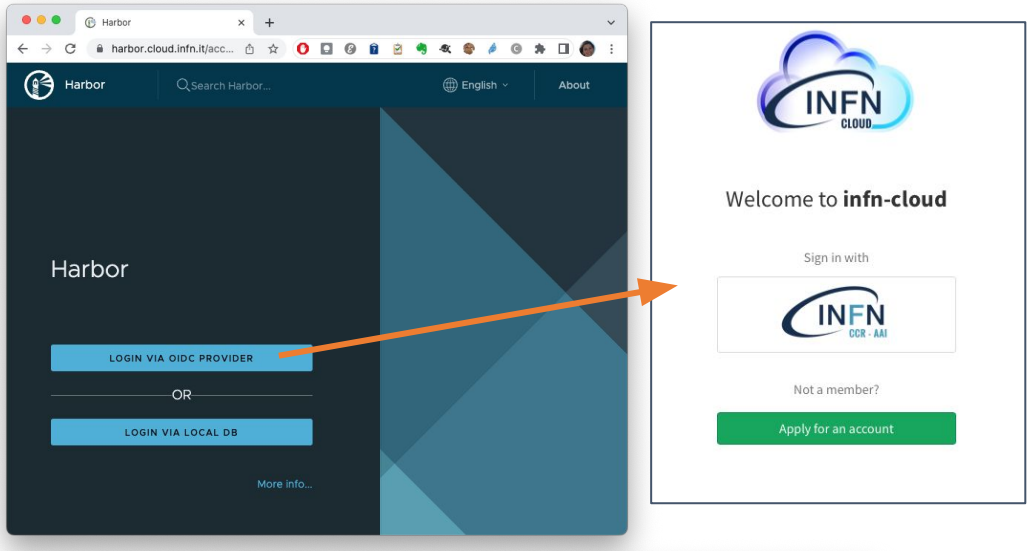
- Virtual machine
- Docker-compose
- Run docker
- Elasticsearch and Kibana
- Kubernetes cluster
- Spark + Jupyter cluster
- HTCondor cluster
- Jupyter with persistence for Notebooks
- Computational enviroment for Machine Learning INFN (ML_INFN)
- Working Station for CYGNO experiment
- Sync&Share aaS

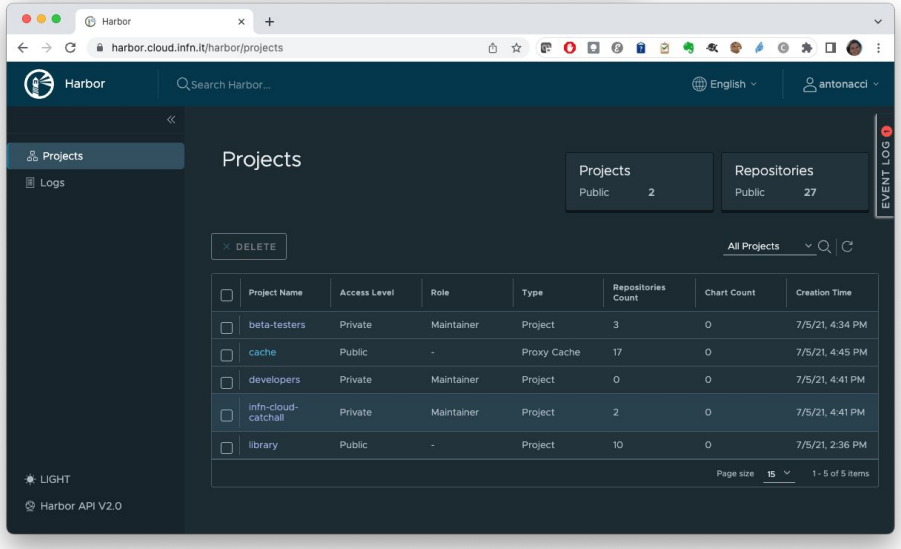
Self-Managed Services

Docker related services

How to manage and deploy containers on INFN Cloud

Harbor: docker registry





Two types of projects supported:

- **Public**: any user can pull images from this project (this is a convenient way to share repositories);
- **Private**: only users who are members of the project can pull images.

Proxy cache configured: when a pull request comes to a proxy cache project, if the image is not cached, Harbor pulls the image from the target registry and serves the pull command as if it is a local image from the proxy cache project.

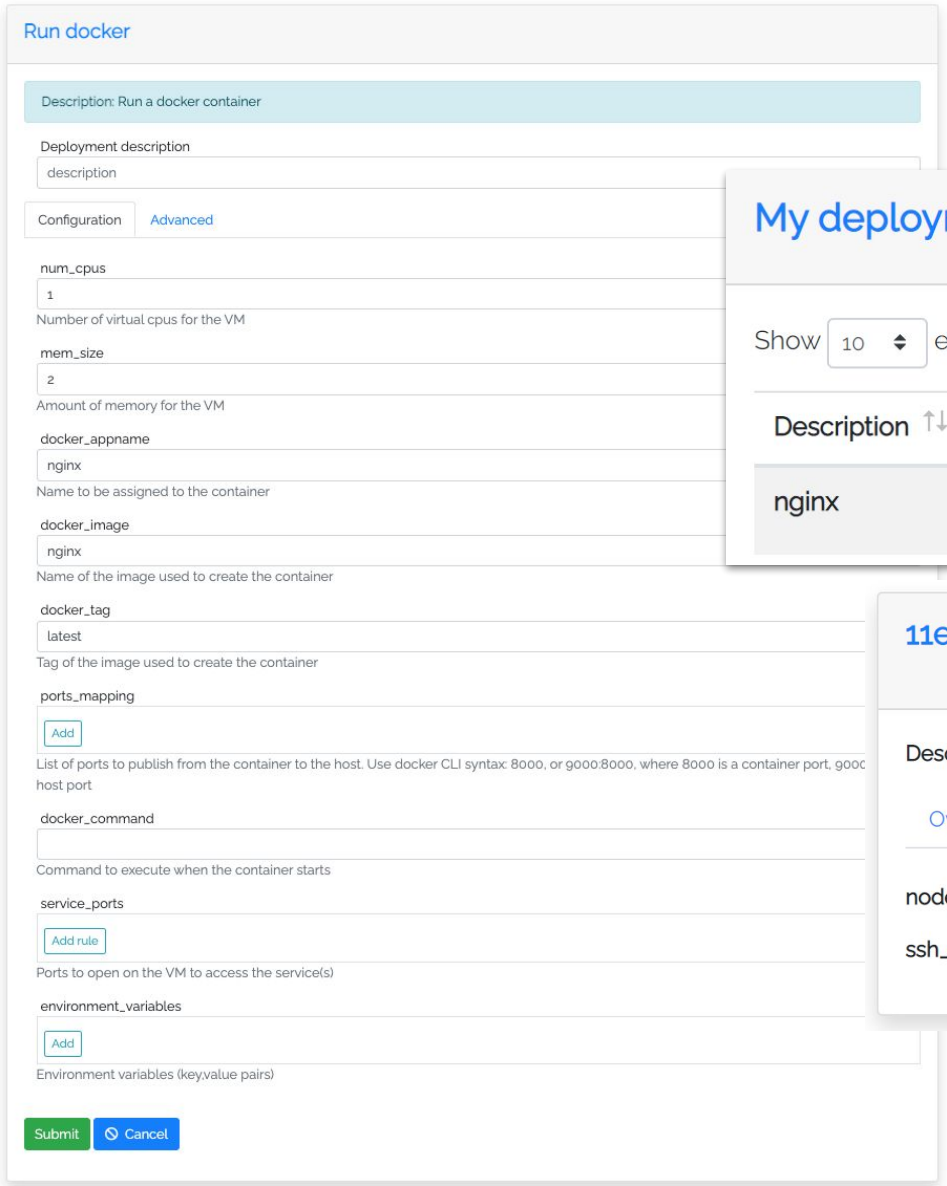
https://harbor.cloud.infn.it

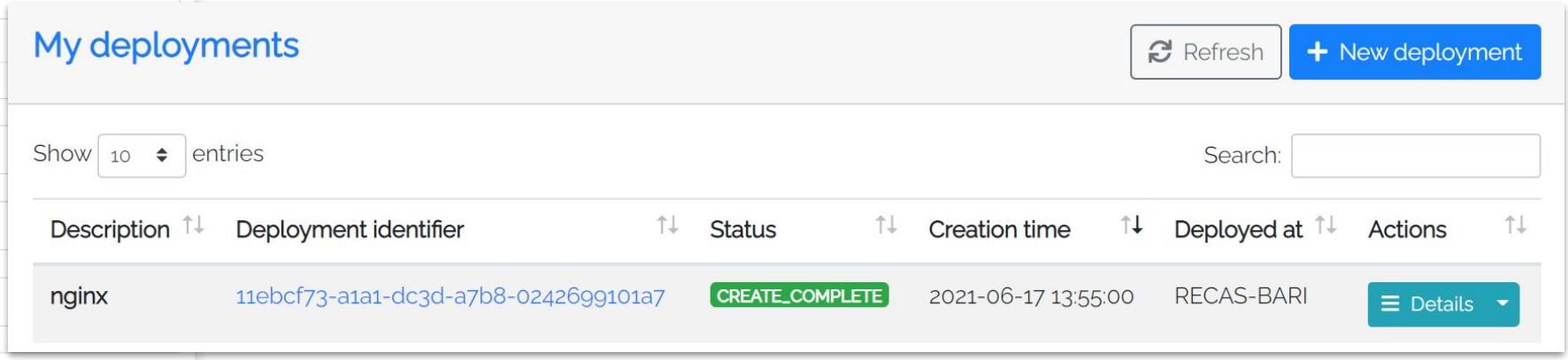
Docker run use-case

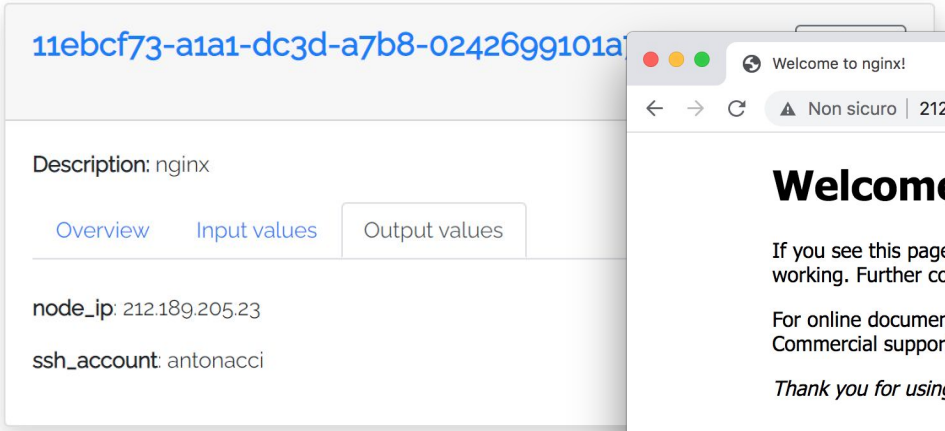
How to run a container on INFN Cloud

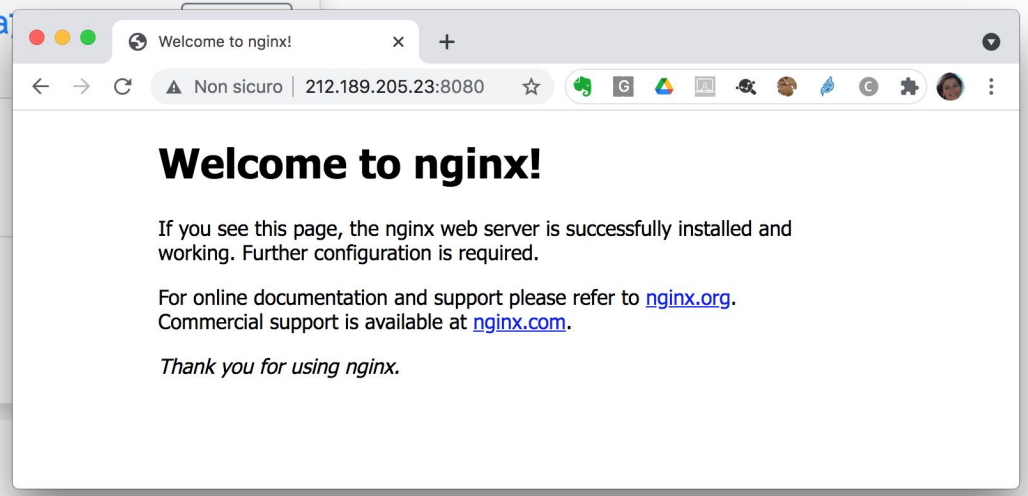
Configure your dockerized service

The configuration form allows you to customize your deployment.

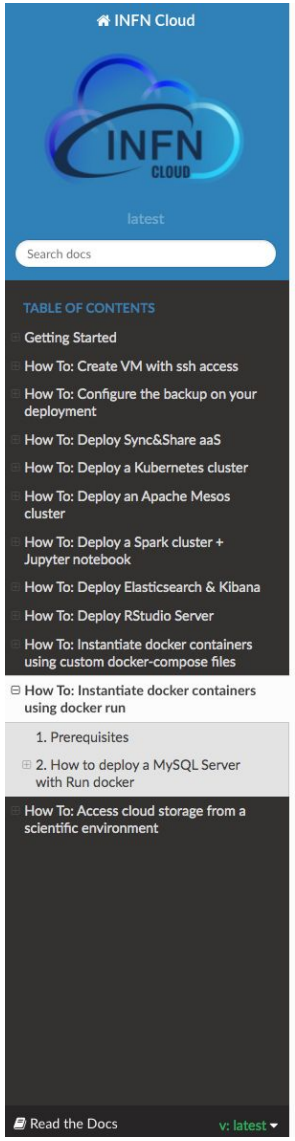


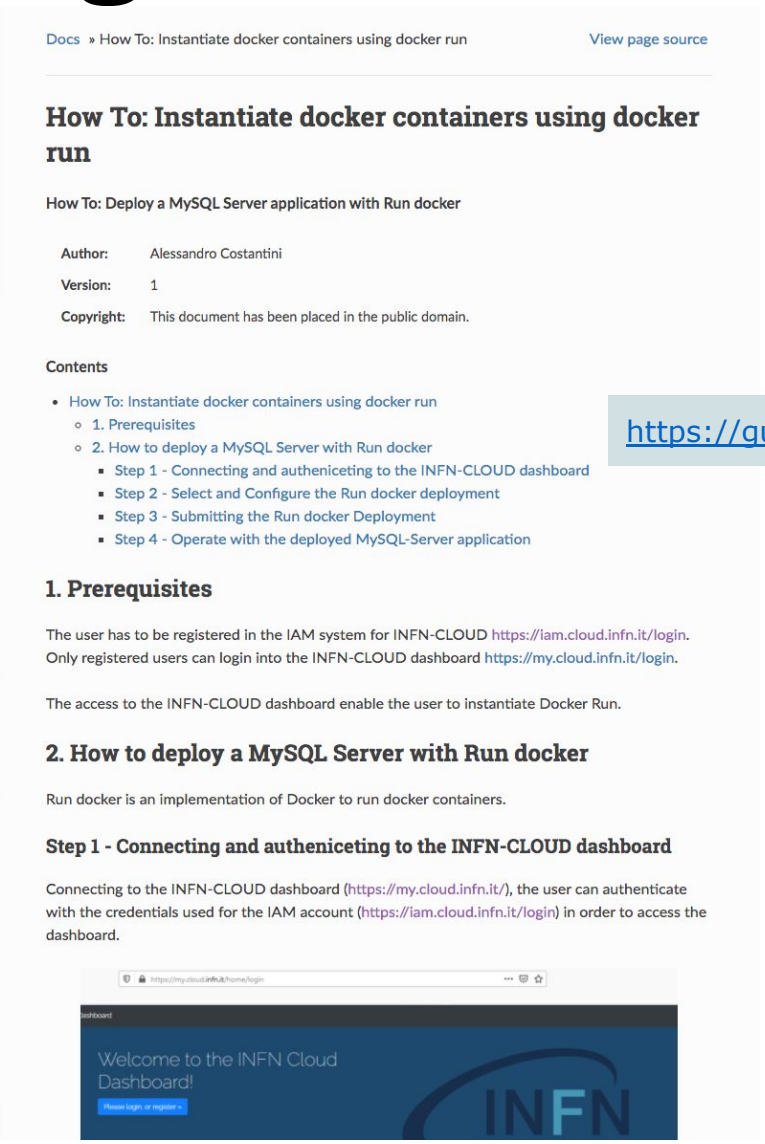




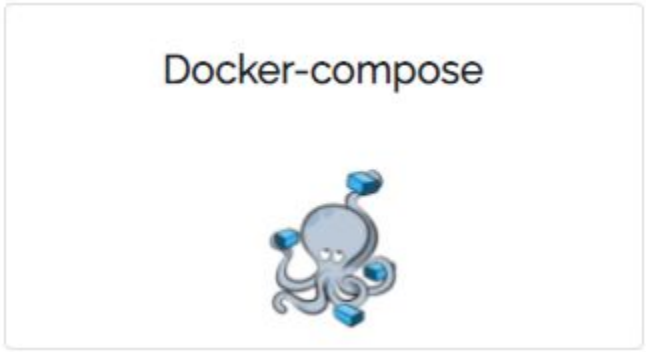


How to su guides.cloud.infn.it





https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto8.html

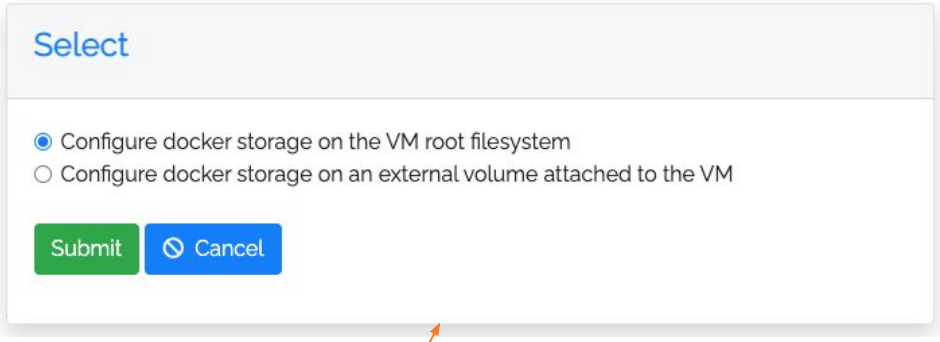


Docker-compose use-case

How to deploy a machine with docker compose pre-installed

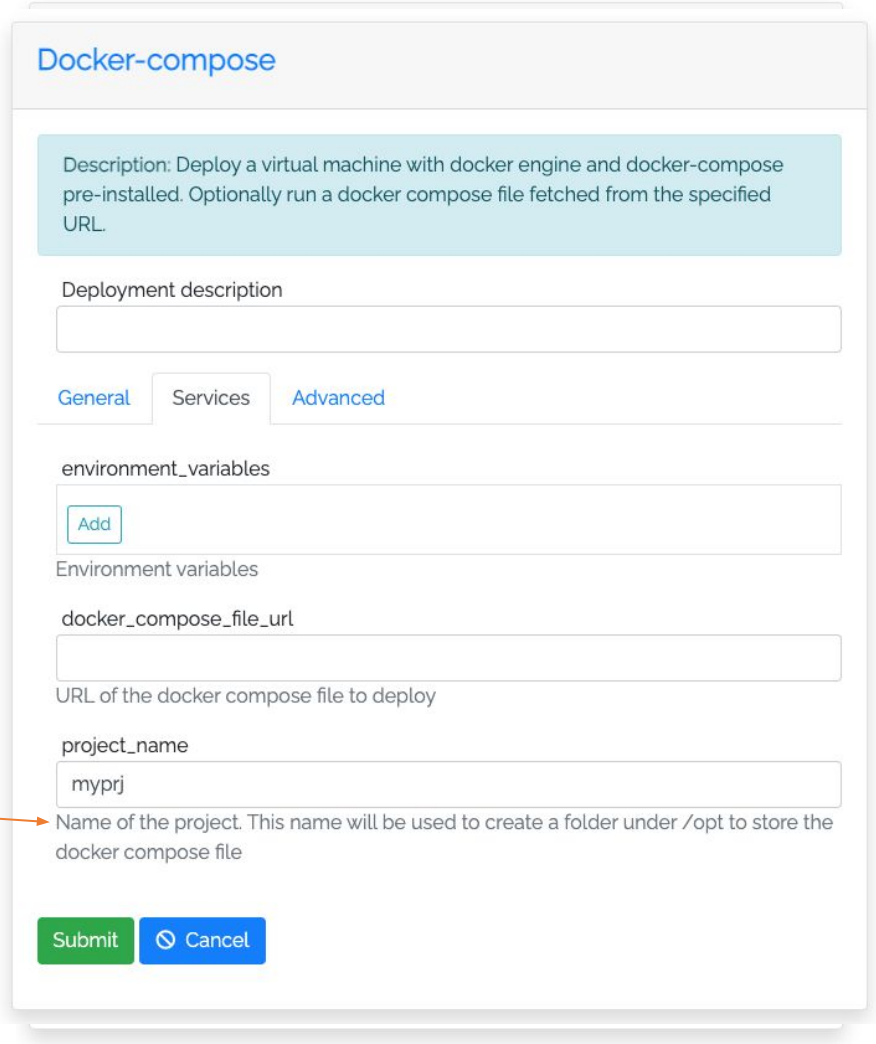
and eventually run a docker-compose file fetched from a given URL

Configure your service

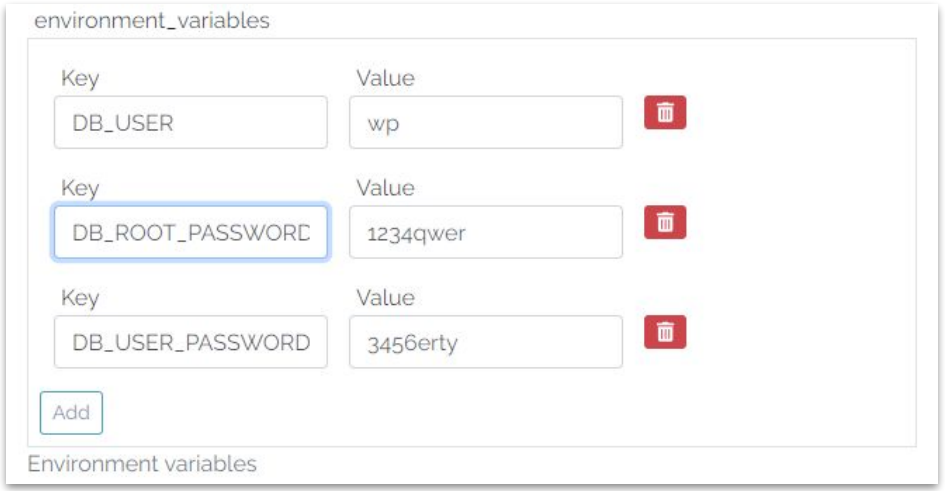


You can choose to

- Put the docker storage on a separate volume
- Configure the machine with only docker and docker-compose or provide a docker compose file URL to start your services



Environment variables management

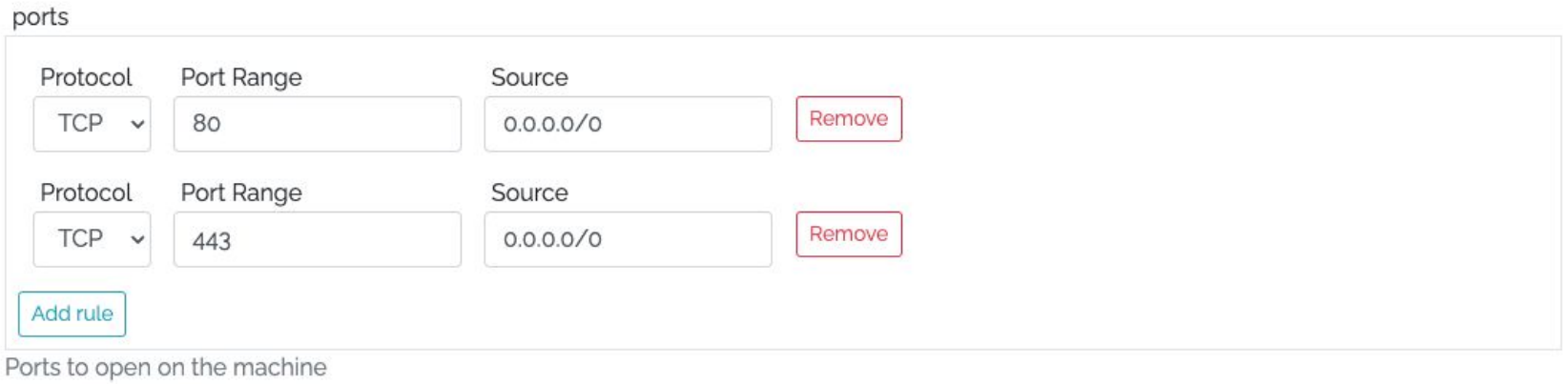


- The special variable *HOST_PUBLIC_IP* is made available by the PaaS system and contains the public IP assigned to the VM
- This env variable can be used as a normal env variable inside the user docker compose file

```
services:
  ........
  app:
    depends_on:
      - db
    image: wordpress
    container_name: app
    volumes:
      - wp-content:/var/www/html/wp-content
    environment:
      - WORDPRESS_DB_HOST=db:3306
      - WORDPRESS_DB_USER=${DB_USER}
      - WORDPRESS_DB_PASSWORD=${DB_USER_PASSWORD}
      - VIRTUAL_HOST=wp.${HOST_PUBLIC_IP}.myip.cloud.infn.it
    expose:
      - 80
```

Ports management

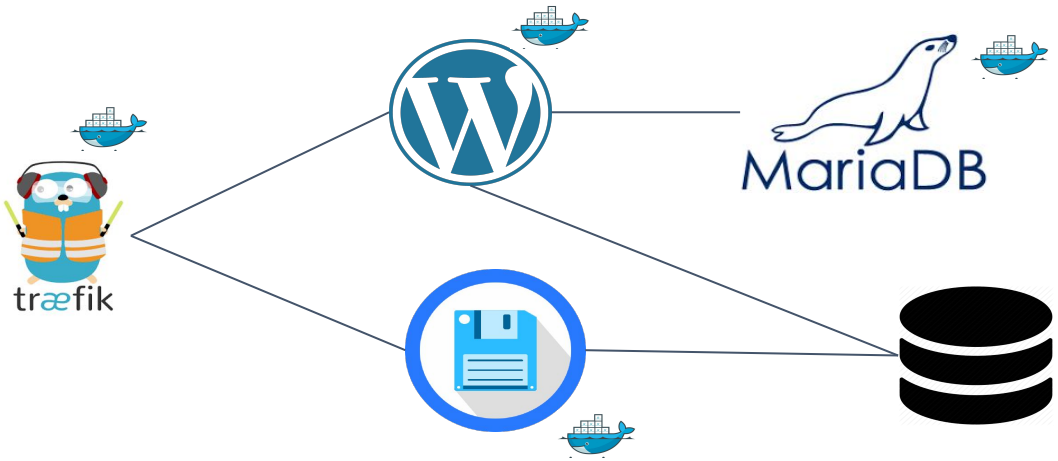
You can define the set of ports that must be automatically opened on the server in order to access your services

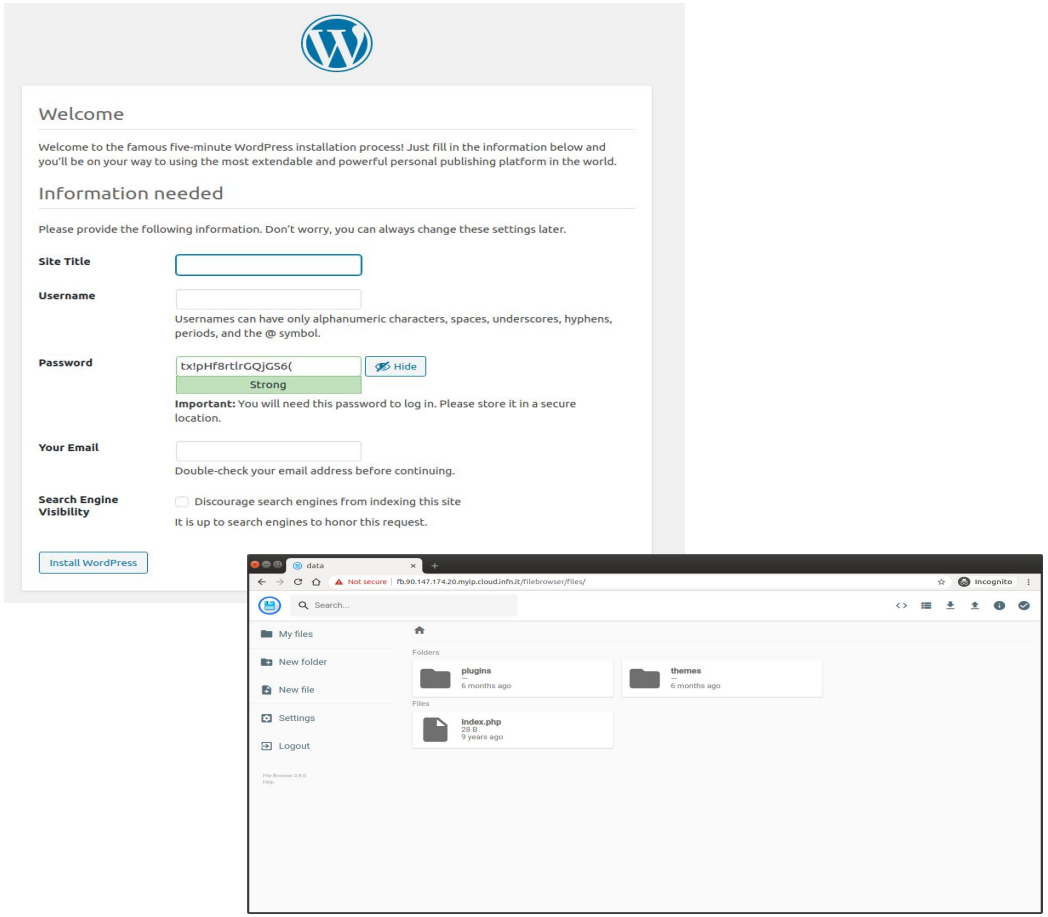


Docker compose example

https://baltig.infn.it/infn-cloud/apps/-/blob/master/compose-example/docker-demo.yaml

Author: Stefano Stalio (LNGS)





DNS @INFN Cloud

INFN Cloud provides a DNSaaS mechanism that associates a DNS name to each VM public IP

$ host **wp.90.147.174.132.myip.cloud.infn.it**
wp.90.147.174.132.myip.cloud.infn.it has address 90.147.174.132

This mechanism is based on xip.io (wildcard DNS) and is exploited for the automatic generation of ssl certificates (e.g. with letsencrypt)

```
services:
 db:
  image: mariadb
  container_name: db
  volumes:
   - db:/var/lib/mysql
  environment:
   - MYSQL_ROOT_PASSWORD=${DB_ROOT_PASSWORD}
   - MYSQL_DATABASE=wordpress
   - MYSQL_USER=${DB_USER}
   - MYSQL_PASSWORD=${DB_USER_PASSWORD}
  expose:
   - 3306
 app:
  depends_on:
   - db
  image: wordpress
  container_name: app
  volumes:
   - wp-content:/var/www/html/wp-content
  environment:
   - WORDPRESS_DB_HOST=db:3306
   - WORDPRESS_DB_USER=${DB_USER}
   - WORDPRESS_DB_PASSWORD=${DB_USER_PASSWORD}
   - VIRTUAL_HOST=wp.${HOST_PUBLIC_IP}.myip.cloud.infn.it
  expose:
   - 80
```

SSL Terminator & Load-balancer

- You can use Traefik as load balancer and SSL terminator. https://traefik.io/traefik/
- Traefik is able to renew letsencrypt certificates

```
services:
  load_balancer:
    image: traefik
    container_name: traefik
    volumes:
      - letsencrypt:/letsencrypt
      - /var/run/docker.sock:/var/run/docker.sock:ro
    ports:
      - "80:80"
      - "443:443"
    command:
      - "--api.insecure=true"
      - "--providers.docker=true"
      - "--providers.docker.exposedbydefault=false"
      - "--entrypoints.web.address=:80"
      - "--entrypoints.websecure.address=:443"
      - "--certificatesresolvers.myhttpchallenge.acme.httpchallenge=true"
      -
"--certificatesresolvers.myhttpchallenge.acme.httpchallenge.entrypoint=w
eb"
      -
"--certificatesresolvers.myhttpchallenge.acme.email=${CONTACT_EMAIL}"
      -
"--certificatesresolvers.myhttpchallenge.acme.storage=/letsencrypt/acme.
json"
```

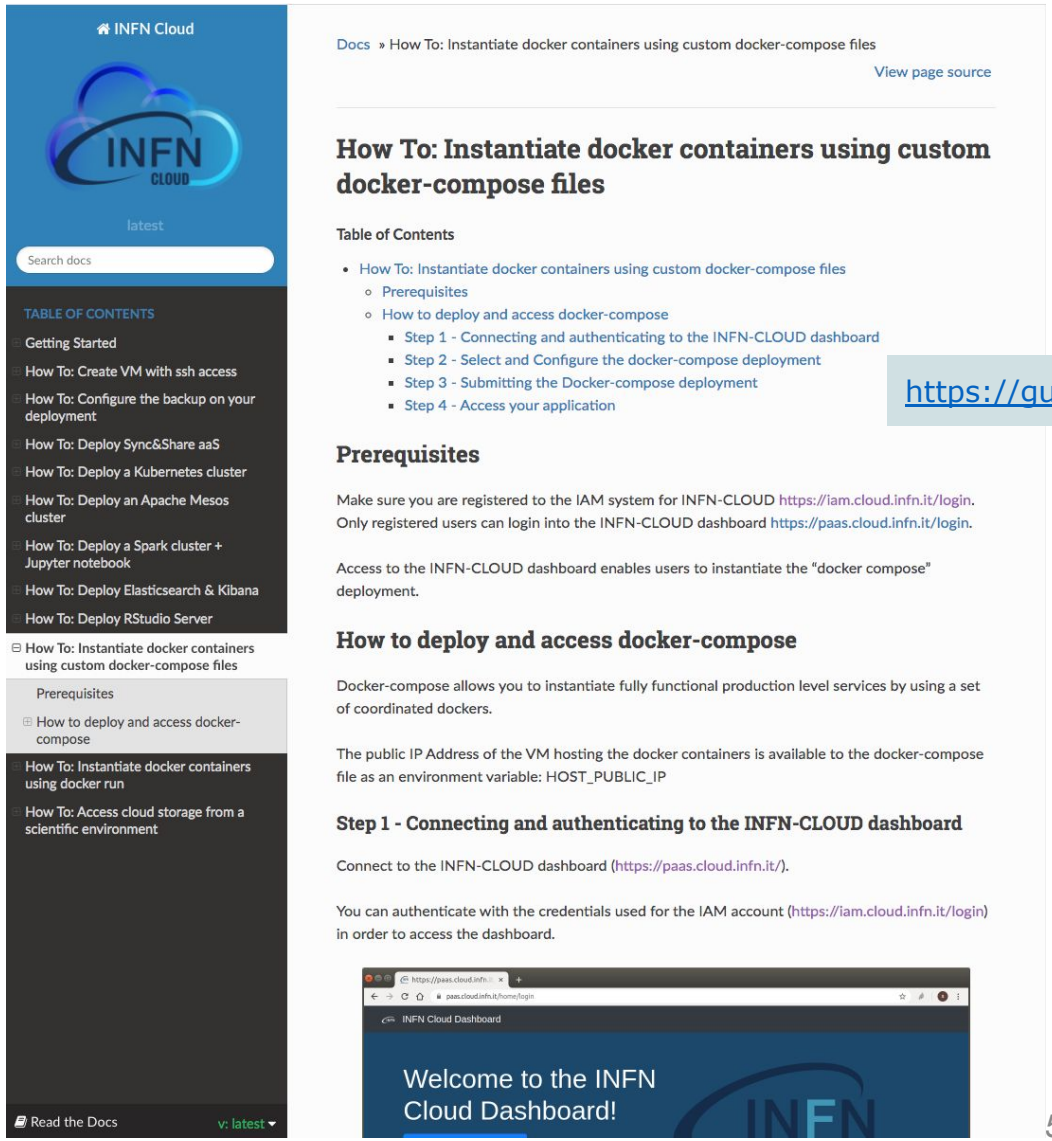
Traefik configuration

Traefik is automatically configured through the labels* exposed by the containers

(*) "A label is a **key=value** *pair that applies metadata to a container."*

```
services:
  app:
   depends_on:
     - db
   image: wordpress
   container_name: app
   volumes:
     - wp-content:/var/www/html/wp-content
   environment:
     - WORDPRESS_DB_HOST=db:3306
     - WORDPRESS_DB_USER=${DB_USER}
     - WORDPRESS_DB_PASSWORD=${DB_USER_PASSWORD}
     - VIRTUAL_HOST=wp.${HOST_PUBLIC_IP}.myip.cloud.infn.it
   expose:
     - 80
   labels:
     - "traefik.enable=true"
     - "traefik.http.middlewares.app-redirect-ssl.redirectscheme.scheme=https"
     - "traefik.http.routers.app-nossl.middlewares=app-redirect-ssl"
     -
"traefik.http.routers.app-nossl.rule=Host(`wp.${HOST_PUBLIC_IP}.myip.cloud.infn.
it`)"
     - "traefik.http.routers.app-nossl.entrypoints=web"
     -
"traefik.http.routers.app.rule=Host(`wp.${HOST_PUBLIC_IP}.myip.cloud.infn.it`)"
     - "traefik.http.routers.app.entrypoints=websecure"
     - "traefik.http.routers.app.tls.certresolver=myhttpchallenge"
     - "traefik.http.routers.app.tls=true"
```

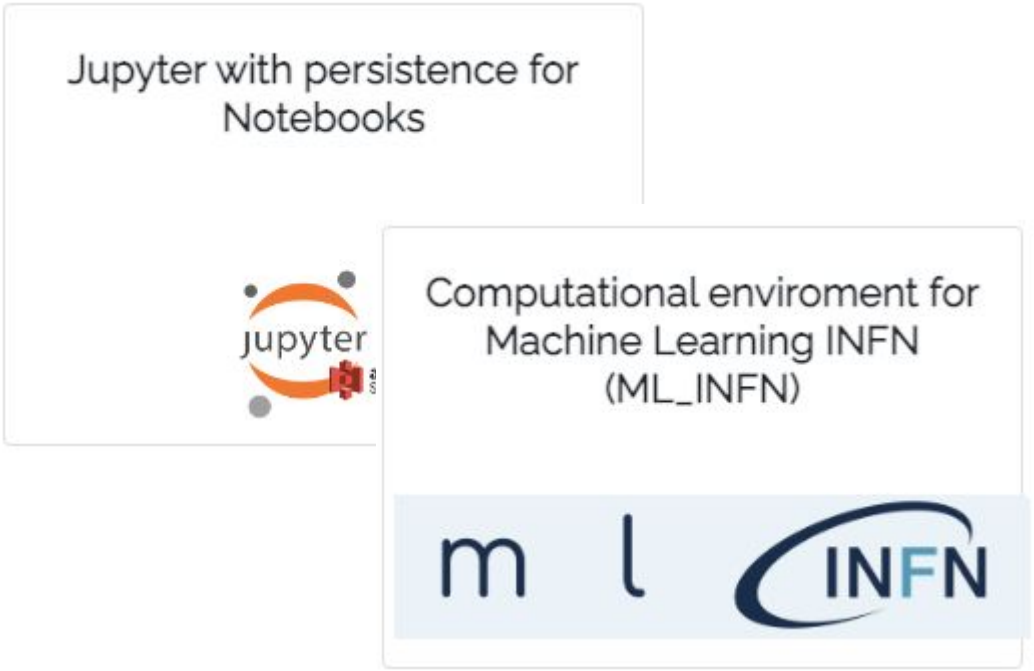
How to su guides.cloud.infn.it



https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto7.html

Docker-based Advanced use-cases:

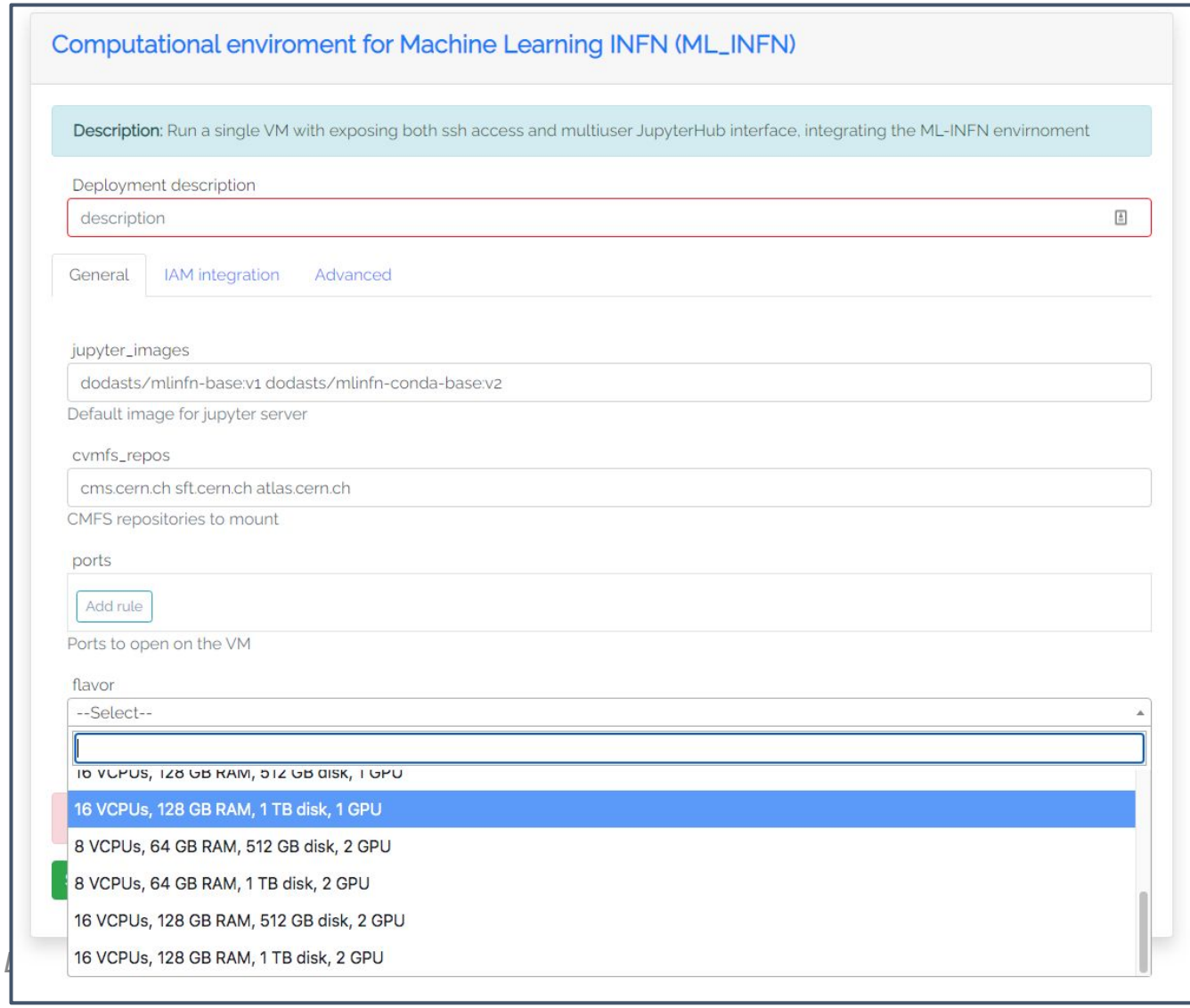
Multi-users JupyterHub
With Persistent storage
With access to GPUs
....



If you are authorized … you can create your own machine!

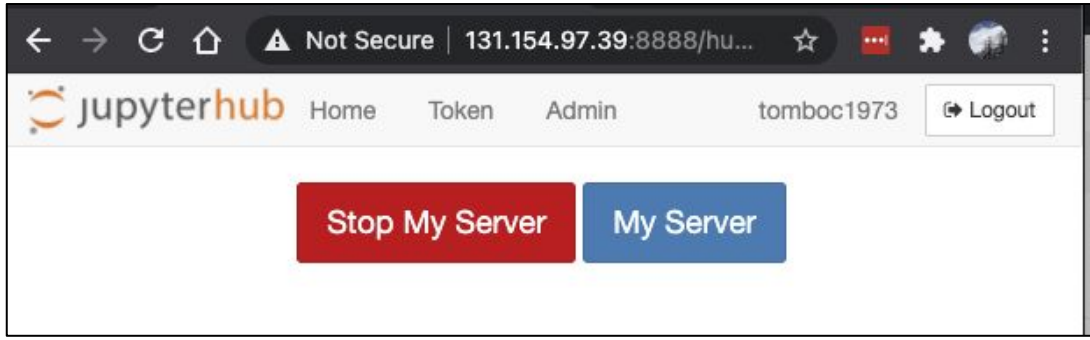
Simple high-level configuration template to create your personal environment

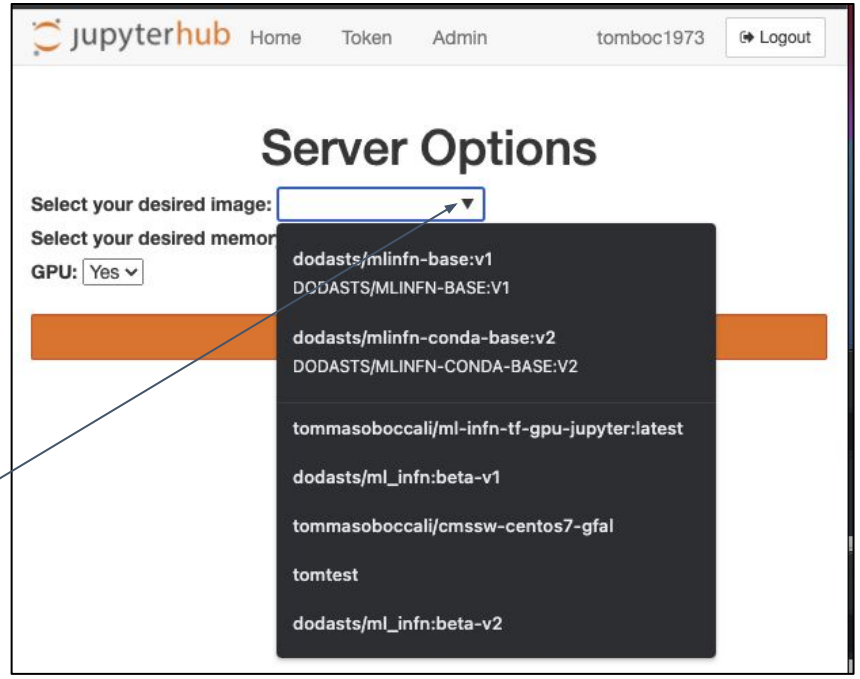
- Either for single user and multi users (group activities)
  - Authorization based on IAM groups
- Ask for CVMFS areas, GPUs, ...



What is inside the VM?

- A **jupyterhub** runs in the VM, and allows **authorized users** to create their running instance through a container (taken either locally, or directly from dockerhub)
- All these containers use the resources of the VM, which are then shared for the user group
- Containers are accessible both via Jupyter Notebooks and via terminal (for the moment via browser)
- The administrator (owner of the service) can access the VM both ssh and via browser



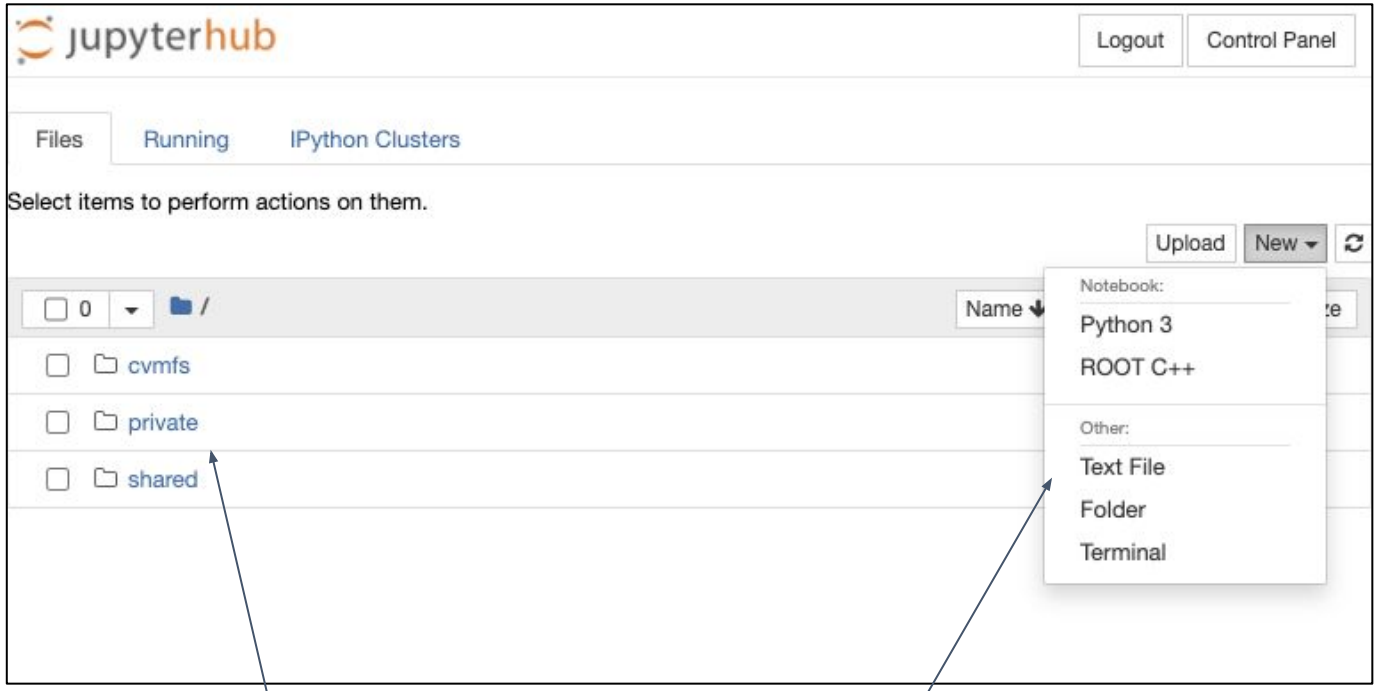


Here you can specify your image

How it is made:

```
root@vnode-0:/home/spiga# docker ps
CONTAINER ID   IMAGE                                    COMMAND                  CREATED          STATUS          PORTS
5db9d94a74d4   dodasts/mlinfn-base:v5                   "jupyterhub-singleus…"   7 seconds ago    Up 5 seconds    8889/tcp
afca0e19e556   grafana/grafana:latest                   "/run.sh -config /op…"   11 days ago      Up 11 days      0.0.0.0:3000->3
6bead4f067ee   prom/prometheus:latest                   "/bin/prometheus --c…"   11 days ago      Up 11 days      0.0.0.0:9090->9
535a161758c6   prom/node-exporter:latest                "/bin/node_exporter"     11 days ago      Up 11 days      9100/tcp
c273ae81940c   google/cadvisor:latest                   "/usr/bin/cadvisor -…"   11 days ago      Up 11 days      8080/tcp
dc53b271c64d   jupyterhub_jupyterhub                    "/usr/bin/python3 /u…"   11 days ago      Up 11 days      8000/tcp
9a120b5bc7cd   jupyterhub_collab_proxy                  "python3 collab_prox…"   11 days ago      Up 11 days      0.0.0.0:8099->8
18cc7311bf14   mircot/jupyterlab_collaborative:ml_base  "jupyter lab --ip=0.…"   11 days ago      Up 11 days      0.0.0.0:8889->8
e0f479af4a86   jupyterhub_backup_service                "cron -f"                11 days ago      Up 11 days
db642fee83e3   jupyterhub/configurable-http-proxy       "/srv/configurable-h…"   11 days ago      Up 11 days      0.0.0.0:8001->8
root@vnode-0:/home/spiga#
```

Access as “User”



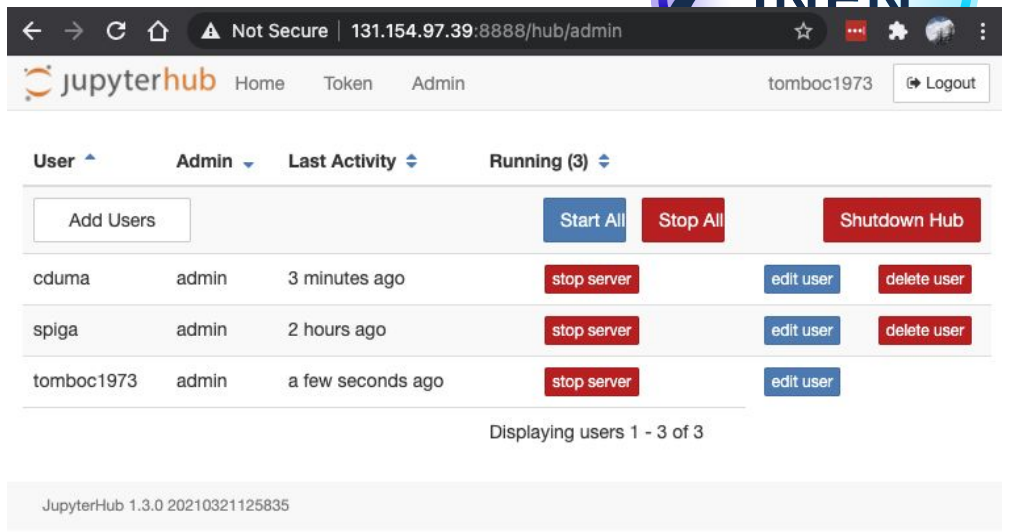
Areas “cvmfs” and “shared” are shared with all the users of the VM

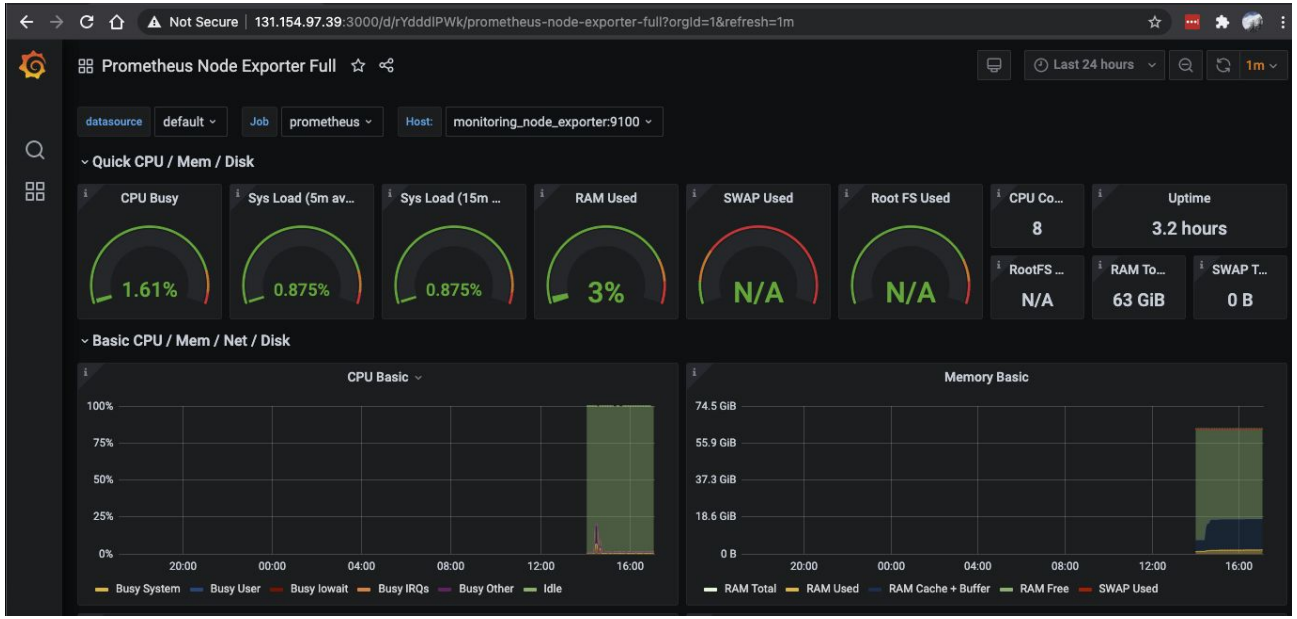
Access granted via notebooks and via terminal

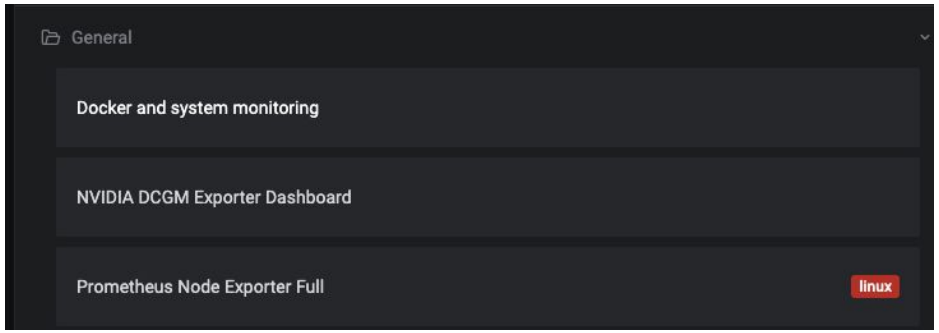
Root access, 2 GPUs available

Monitoring etc

- The administrator can manage containers
- All users can see detailed monitoring information



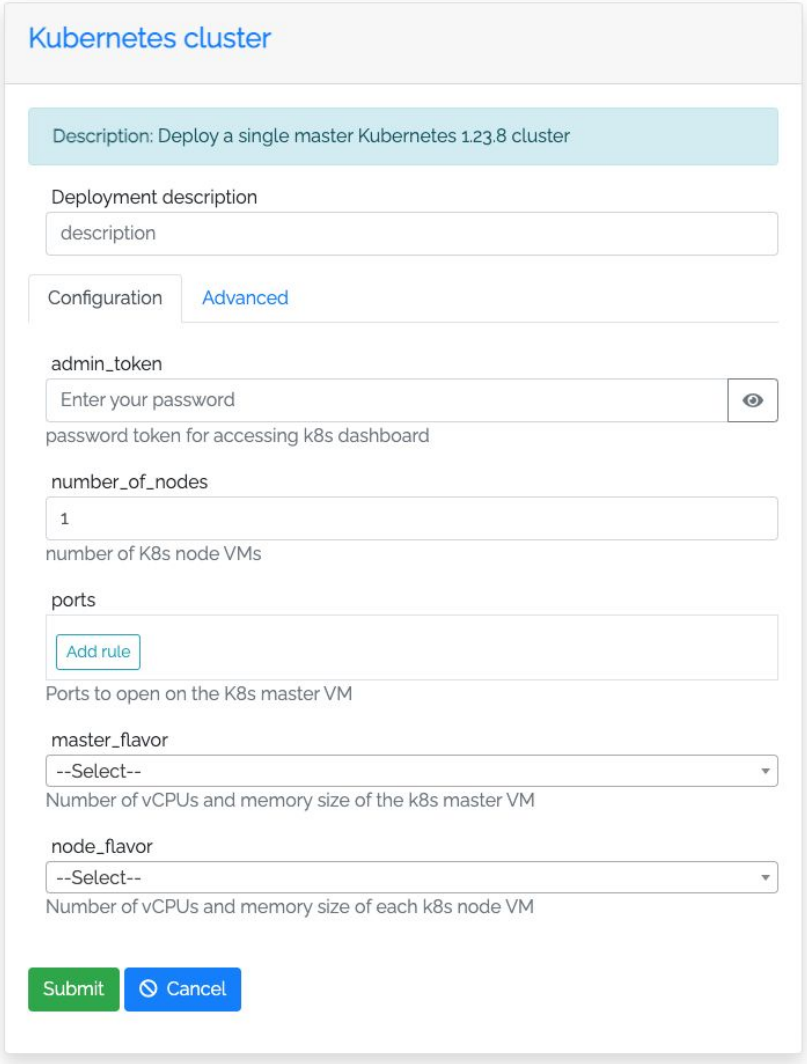




Kubernetes cluster use-case

How to deploy a complete k8s cluster on INFN Cloud

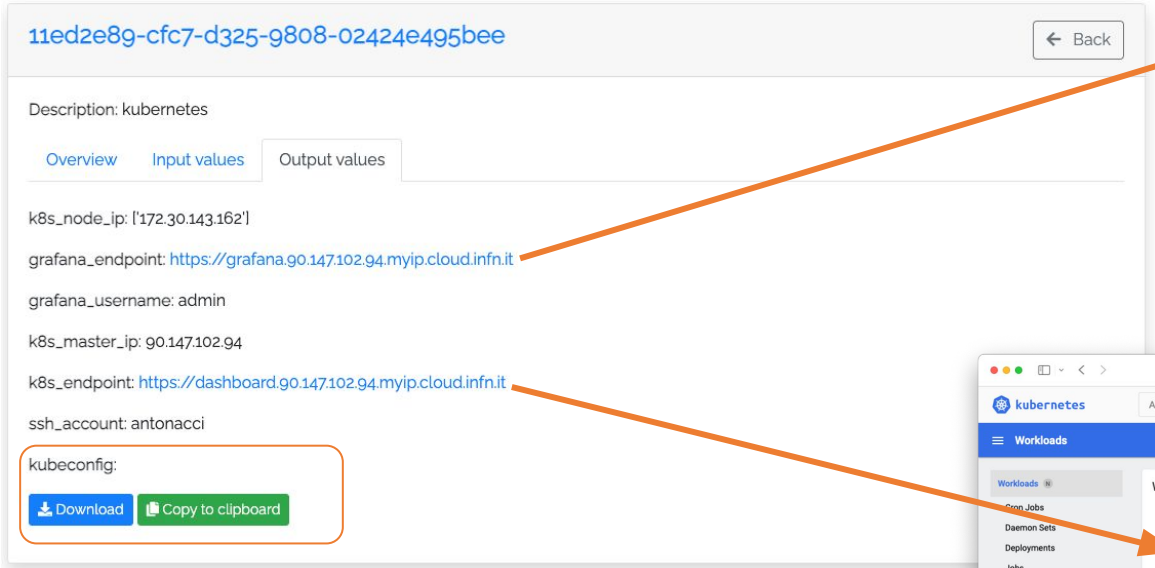
Configure your cluster

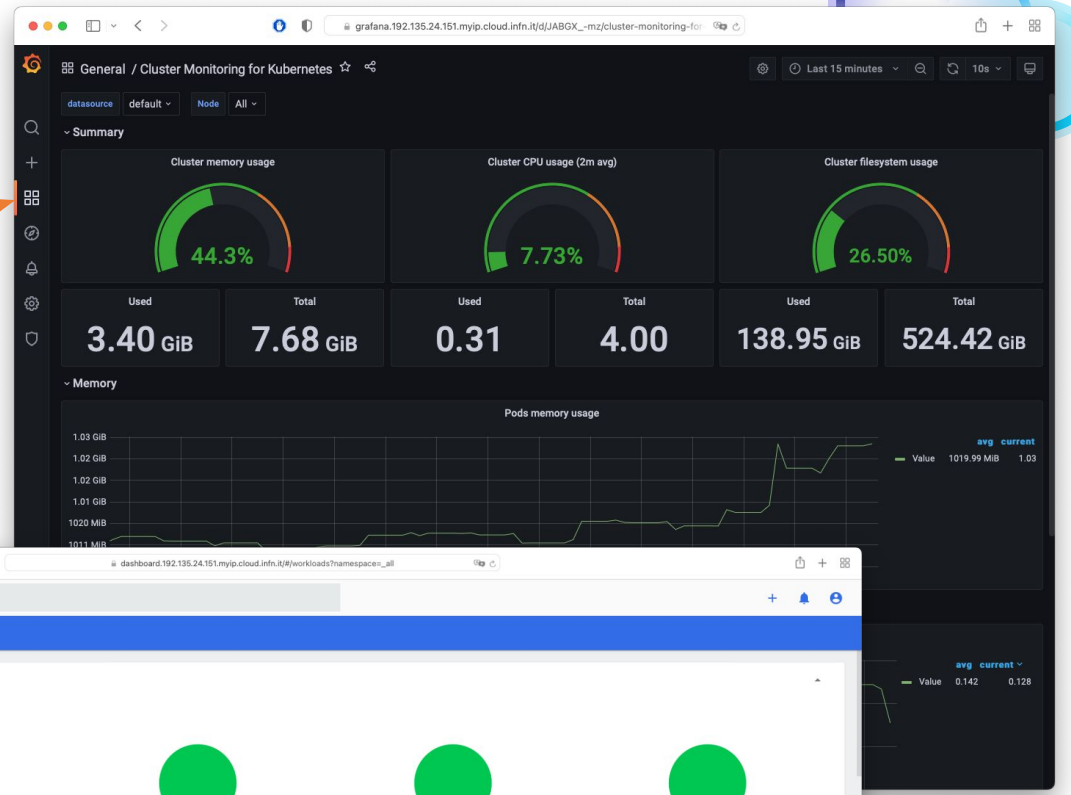


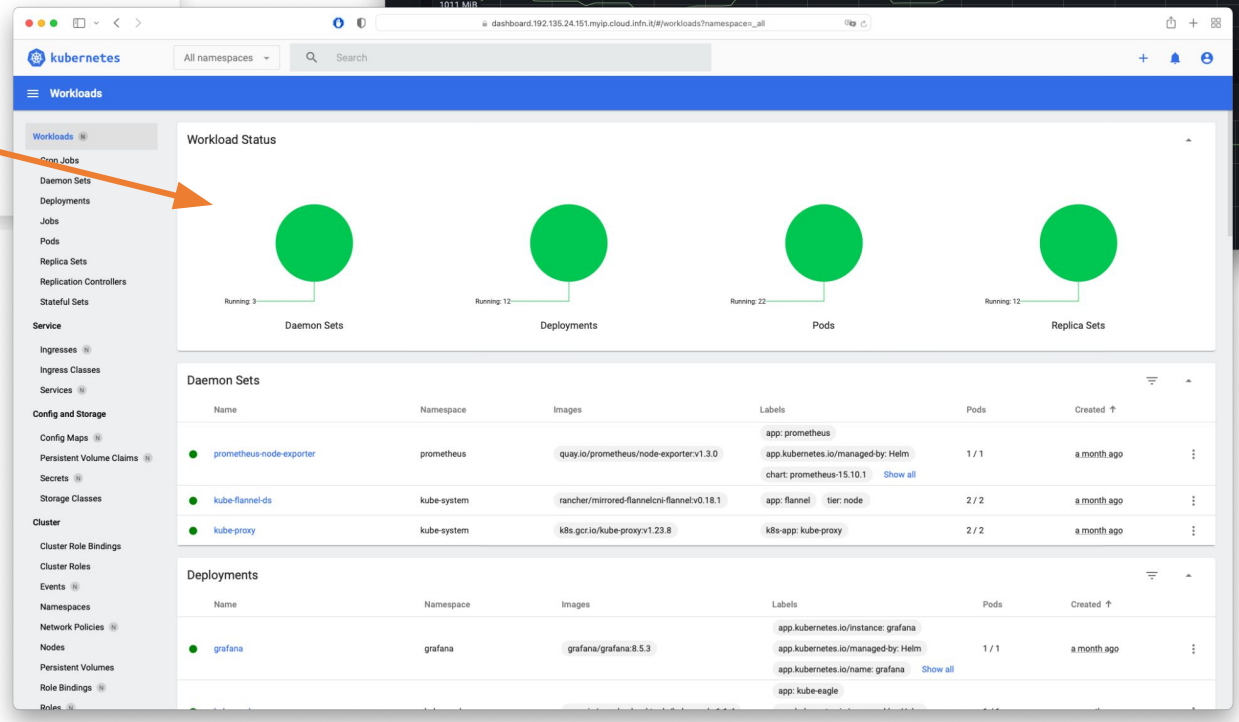
The configuration form allows you to customize your cluster:

- Number of nodes
- Ports to be opened on the master node
- Flavor for the master and node servers

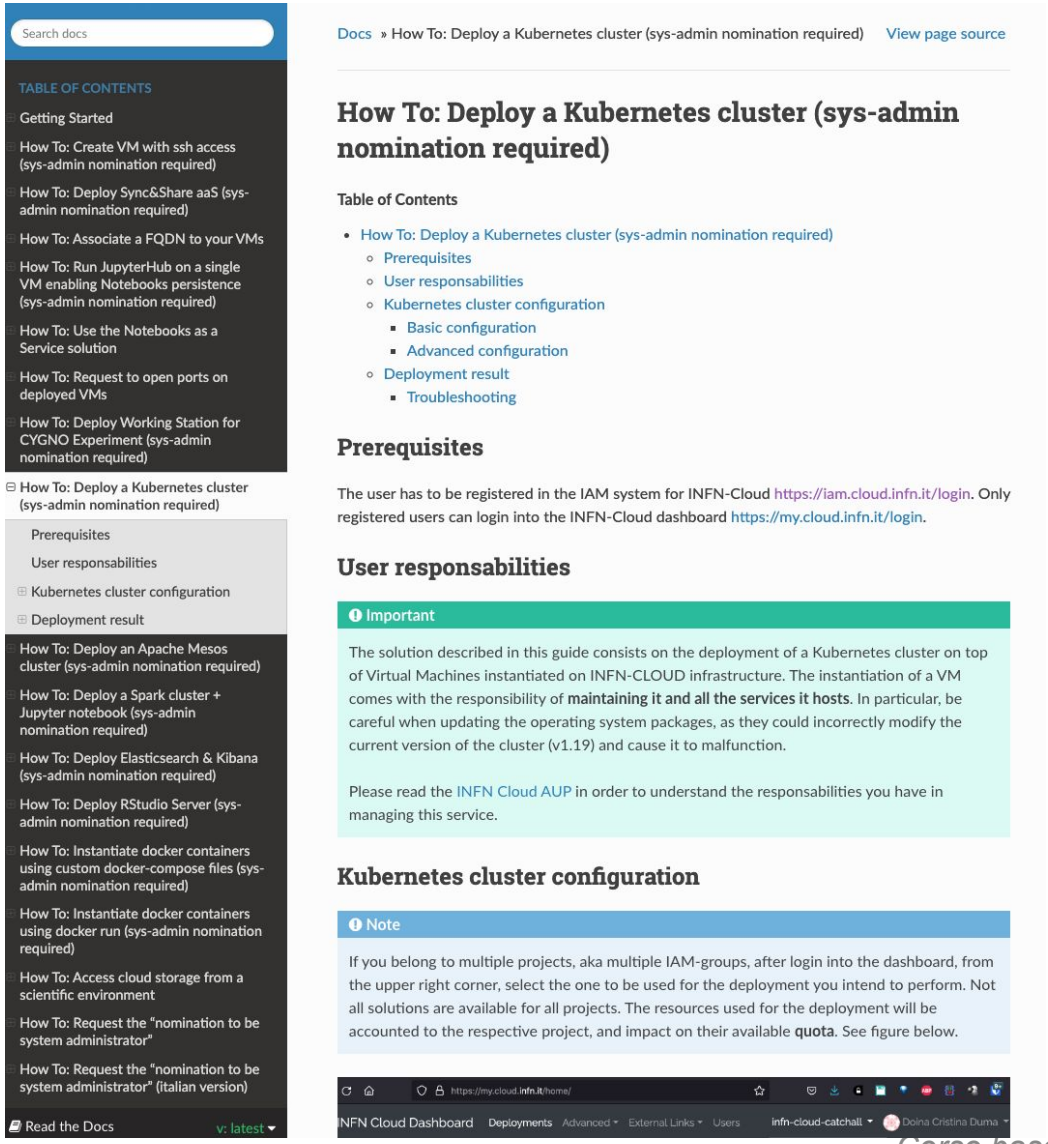
Access your services







How to su guides.cloud.infn.it



https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto2.html

Advanced k8s-based services

Jupyter + Spark + K8s

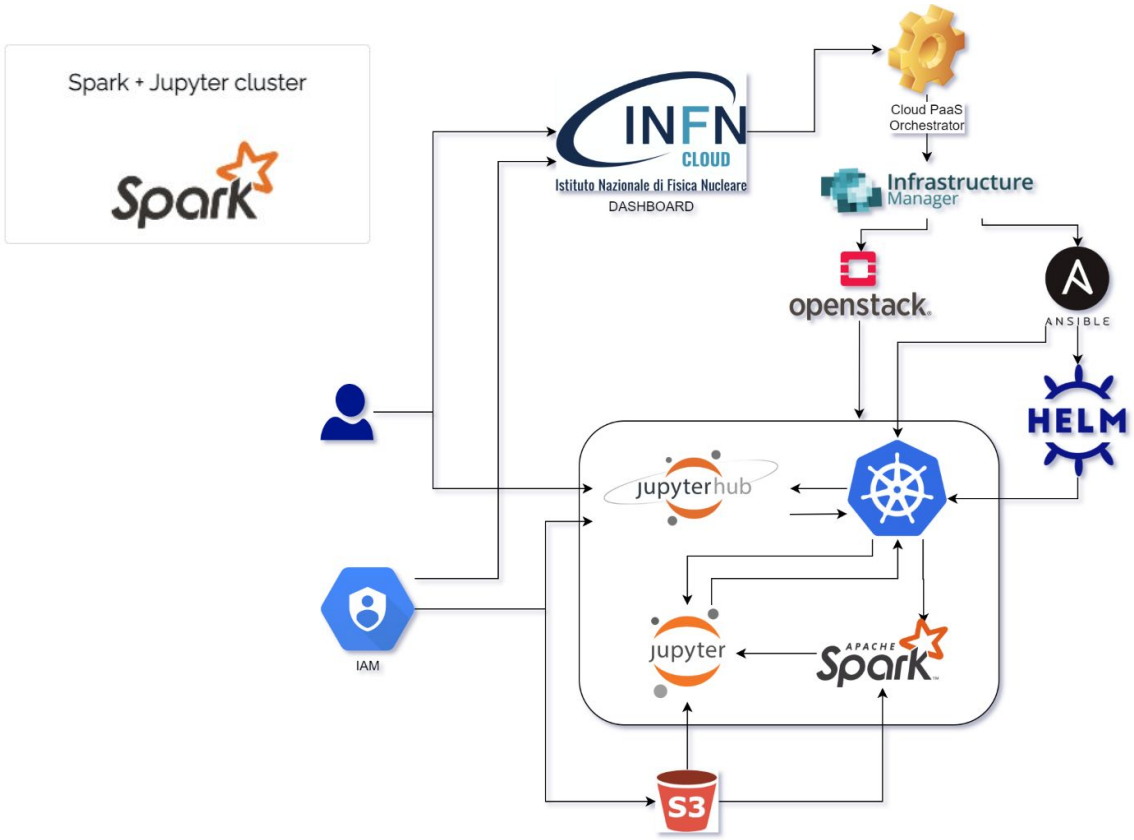


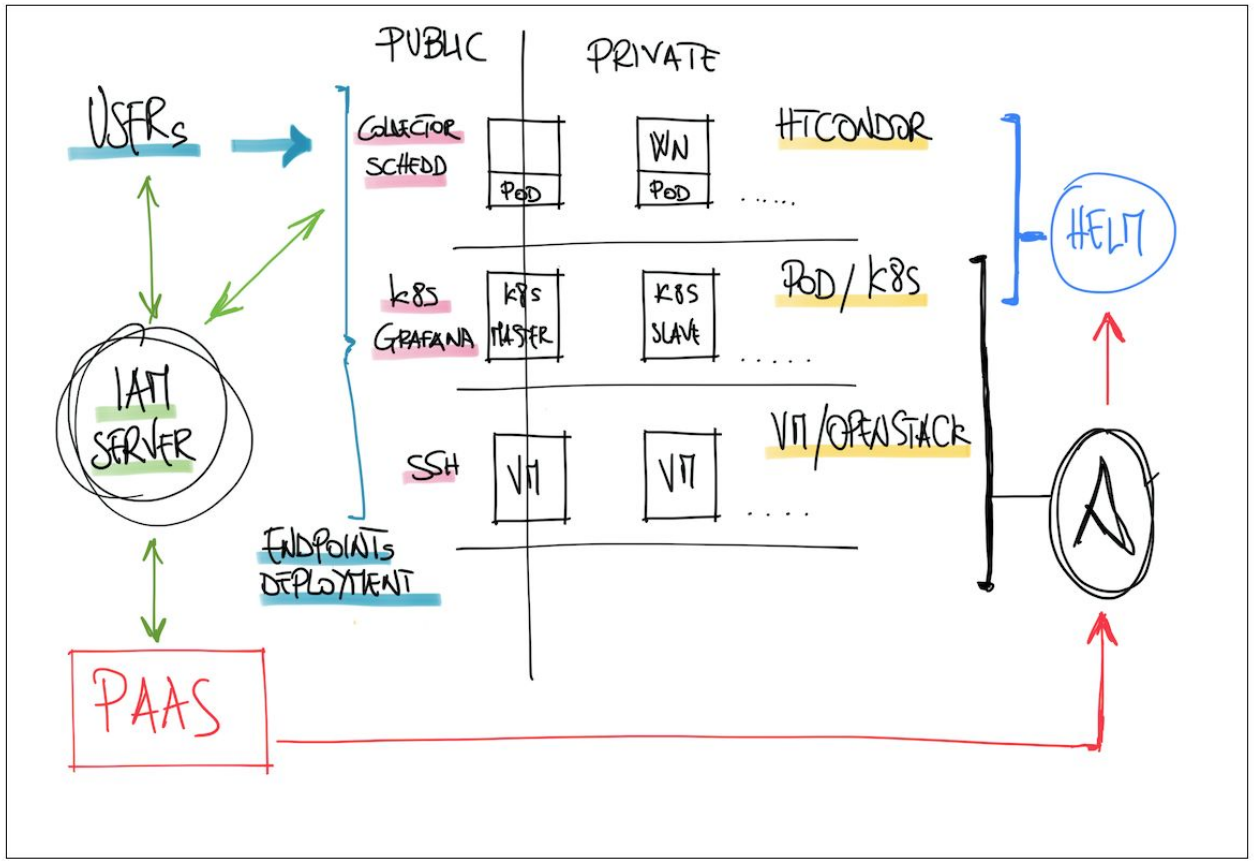
Figure 7: Jupyter notebook example.

In the Spark clusters connection box you can specify the Spark configuration, as shown in fig. 8.

Advanced k8s-based services (2)

HTCondor + K8s

HTCondor cluster



This deployment instantiate a k8s cluster which is then exploited to automatically deploy a working HTCondor cluster.

The HTCondor cluster deployment is composed by three main components, the CCB, the SCHEDD and the WN, each running on a dedicated POD.

Conclusions

The goal of INFN Cloud is to provide end-users with compute and storage services by offering

- a **portfolio of technical solutions** already developed but extensible – continuously evolving following a **user driven development approach**
- technical support for the end user applications migration to a cloud-based environment
- **transparent** solutions hiding the resources allocation complexity in a **federation of distributed clouds**

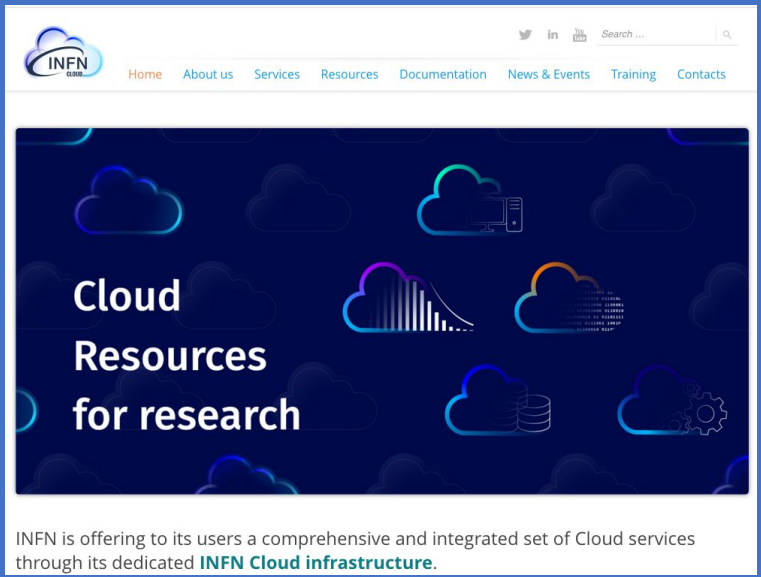
The high-level services shown in this presentation are part of the current portfolio:

- They provide a simple way to run docker containers on cloud resources
- Further (more complex) services have been built starting from these building blocks

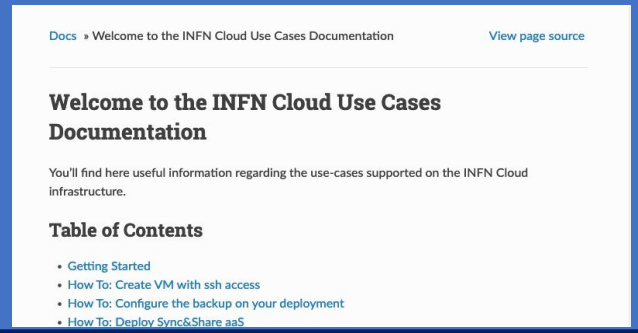
If you want to implement a new service or you need to customize an existing one, please contact us at: **cloud-support@infn.it** and you will be redirected to the proper INFN Cloud support team

References

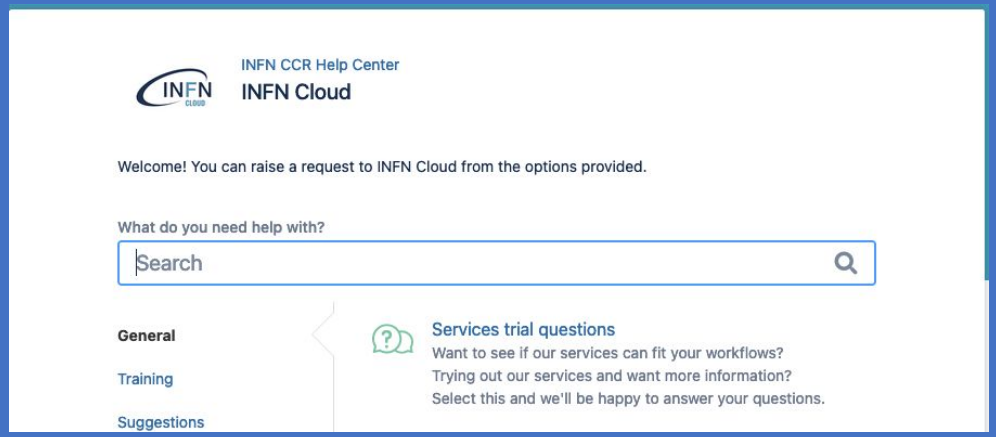
Web site:
https://www.cloud.infn.it



Documentation :
https://guides.cloud.infn.it/docs/users-guides/en/latest/



Support :
https://servicedesk.cloud.infn.it or cloud-support@infn.it



Thank you

for your attention!

www.cloud.infn.it

For general communications email us at **cloud@lists.infn.it**

To ask for support write to our mailing list **cloud-support@infn.it**, integrated with our ServiceDesk