



# Introduction to INFN Cloud services

Marica Antonacci (INFN BA)  
Corso *Big Data e ML*, 7-11 Nov 2022



# 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

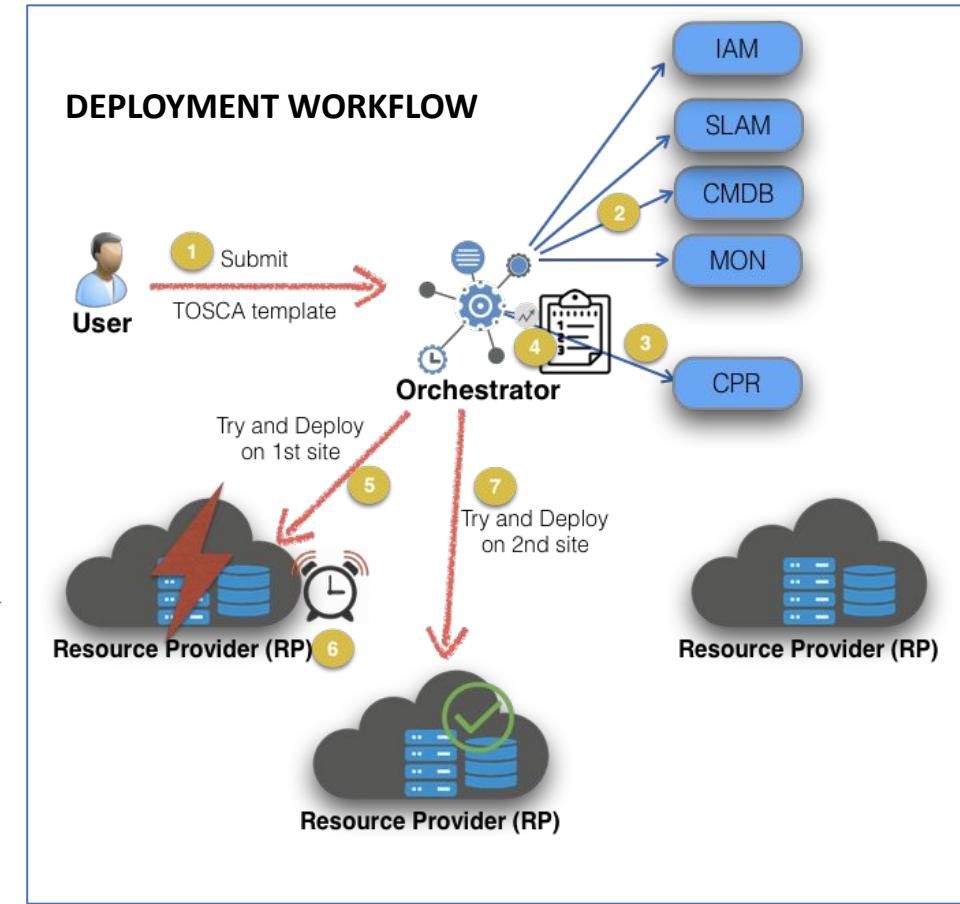
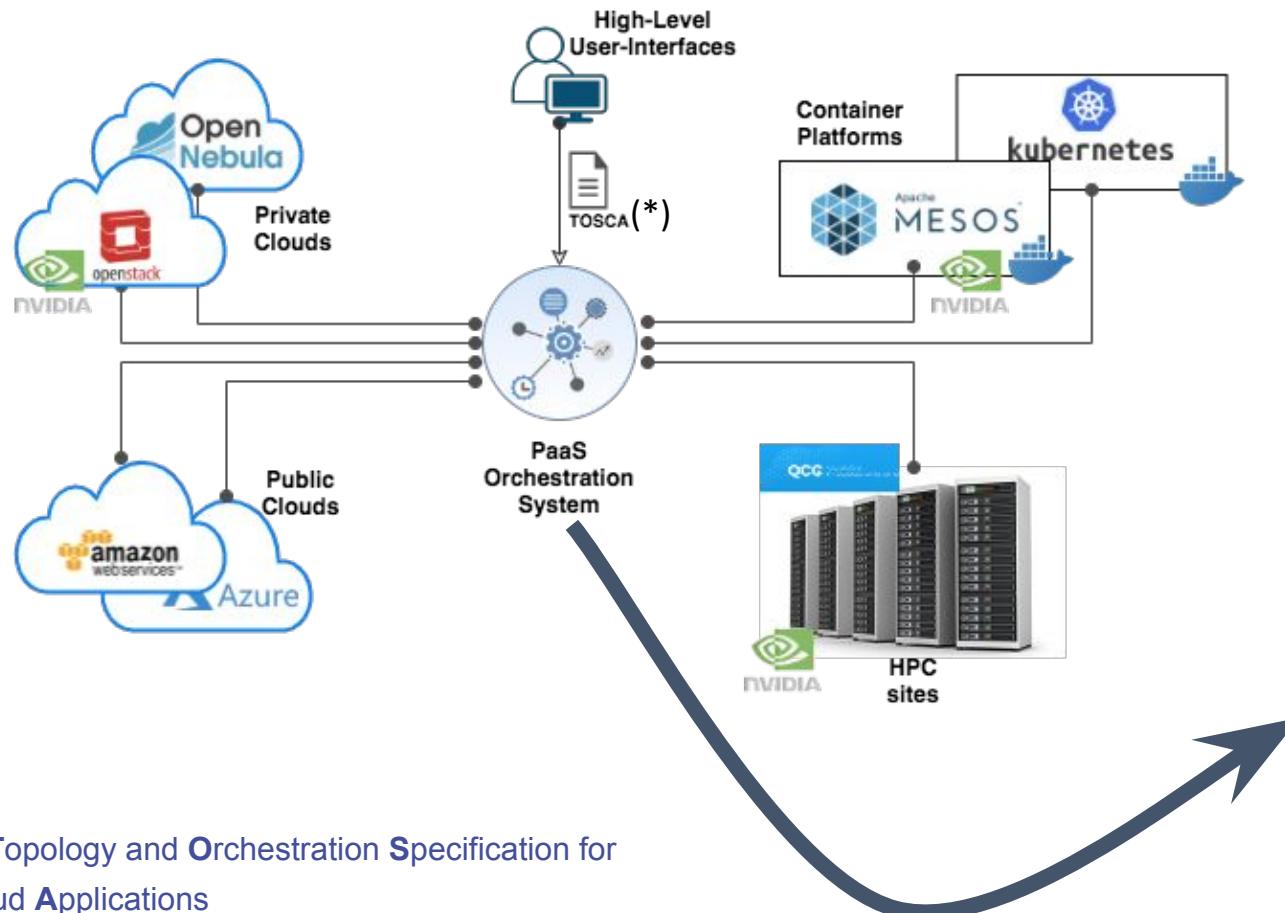
Architecturally INFN Cloud is a **federation** of existing infrastructures

- ❑ **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)



(\*) Topology and Orchestration Specification for Cloud Applications

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**

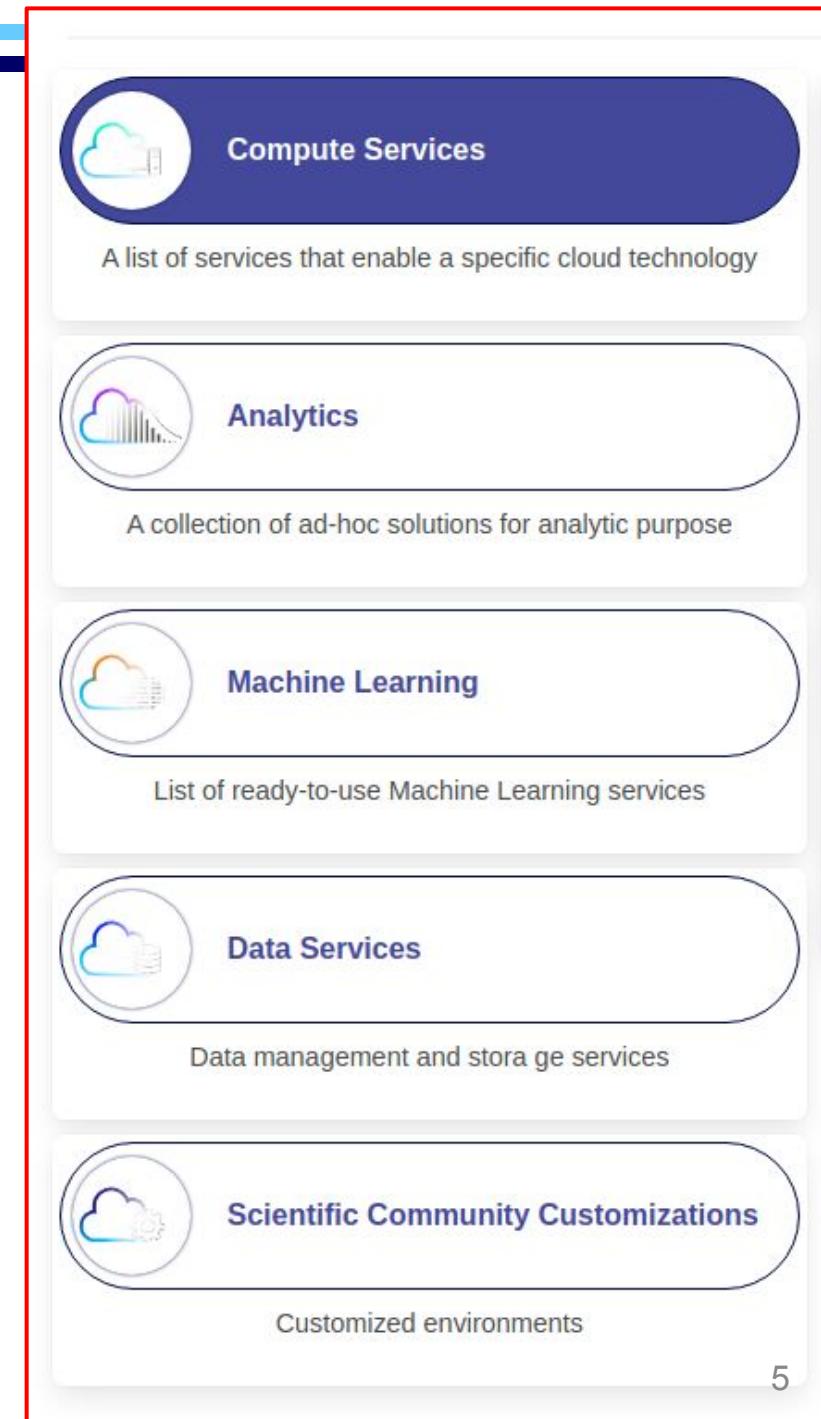
- Spark e/o ElasticSearch e Kibana, R, etc..

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

- Jupyter Notebooks with persistent storage (replicated)

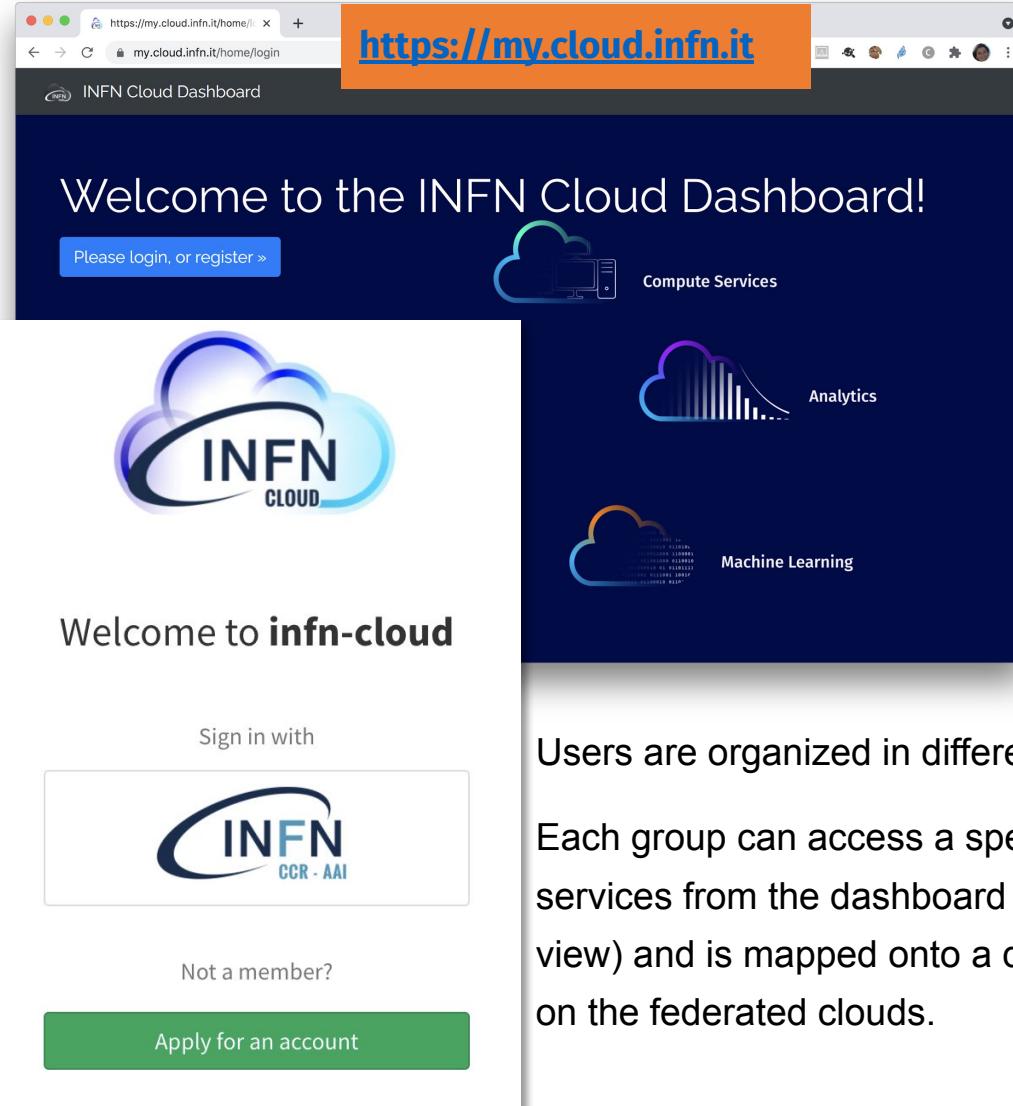
**Dynamic Clusters** even designed and tuned taking into account the specific communities needs;

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



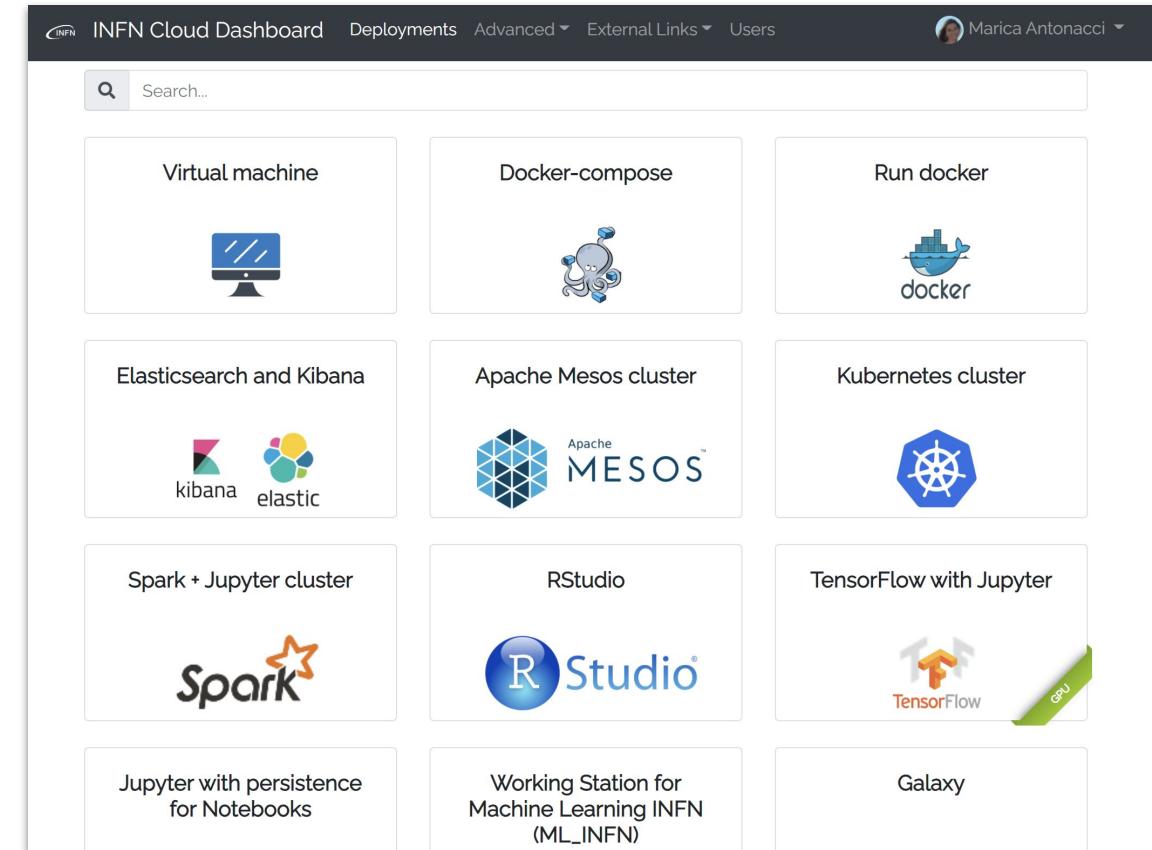


# The INFN Cloud Dashboard



The screenshot shows the INFN Cloud Dashboard homepage. At the top, there's a header with the URL <https://my.cloud.infn.it>. Below the header, a large banner says "Welcome to the INFN Cloud Dashboard!" with a "Please login, or register »" button. On the right side of the banner, there are three service icons: "Compute Services" (cloud with server), "Analytics" (cloud with bar chart), and "Machine Learning" (cloud with binary code). Below the banner, there's a section titled "Welcome to infn-cloud" with "Sign in with" options for INFN CCR-AAI and "Not a member? Apply for an account". A text block states: "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."

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



The screenshot shows the INFN Cloud Dashboard with a grid of service options. The top navigation bar includes "INFN Cloud Dashboard", "Deployments", "Advanced", "External Links", "Users", and a user profile for "Marica Antonacci". The grid contains 12 items:

- Virtual machine (Icon: monitor)
- Docker-compose (Icon: octopus)
- Run docker (Icon: docker)
- Elasticsearch and Kibana (Icons: kibana, elastic)
- Apache Mesos cluster (Icon: MESOS)
- Kubernetes cluster (Icon: helm)
- Spark + Jupyter cluster (Icon: Spark)
- RStudio (Icon: R Studio)
- TensorFlow with Jupyter (Icon: TensorFlow GPU)
- Jupyter with persistence for Notebooks
- Working Station for Machine Learning INFN (ML\_INFN)
- Galaxy

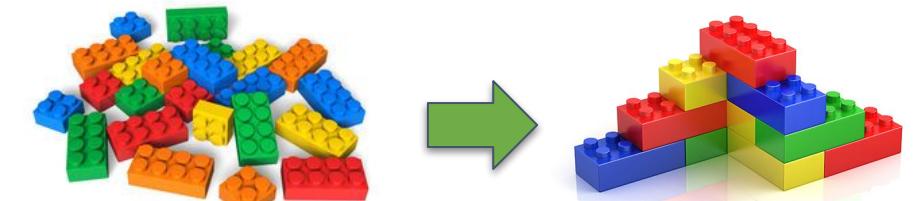
# The service catalogue

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**





# 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 environment for Machine Learning INFN (ML\_INFN)



Working Station for CYGNO experiment



Sync&Share aaaS



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

Add rule

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

Submit

Cancel

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

Submit

Cancel

# Service request customization

**Virtual machine**

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

Deployment description

description

Configuration Advanced

1 service\_ports

Add rule

Ports to open on the host

2 flavor

--Select--

Number of vCPUs and memory size of the Virtual Machine

3 operating\_system

--Select--

Operating System for the Virtual Machine

Submit Cancel

```

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. the ports specification)

service\_ports

Protocol	Port Range	Source	Remove
TCP	80	0.0.0.0/0	Remove
TCP	443	0.0.0.0/0	Remove

Add rule

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



# Available services

[harbor.cloud.infn.it](https://harbor.cloud.infn.it)



HARBOR

[minio.cloud.infn.it](https://minio.cloud.infn.it)



MINIO

[hub.cloud.infn.it](https://hub.cloud.infn.it)



Fully-Managed  
Services

Virtual machine



Elasticsearch and Kibana



Docker-compose



Run docker



HTCondor cluster



Jupyter with persistence for Notebooks



Computational enviroment for Machine Learning INFN (ML\_INFN)



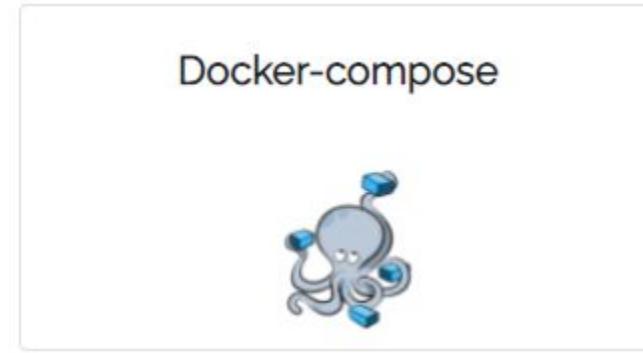
Working Station for CYGNO experiment



Sync&Share aaS



Self-Managed  
Services



# Docker-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

Select

Configure docker storage on the VM root filesystem  
 Configure docker storage on an external volume attached to the VM

**Submit** **Cancel**

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

Docker-compose

Description: Deploy a virtual machine with docker engine and docker-compose pre-installed. Optionally run a docker compose file fetched from the specified URL.

Deployment description

General Services Advanced

environment\_variables

Add

Environment variables

docker\_compose\_file\_url

URL of the docker compose file to deploy

project\_name

myprj

Name of the project. This name will be used to create a folder under /opt to store the docker compose file

**Submit** **Cancel**

# Environment variables management

environment\_variables

Key	Value	
DB_USER	wp	
DB_ROOT_PASSWORD	1234qwer	
DB_USER_PASSWORD	3456erty	

**Add**

Environment variables

- 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

ports

Protocol	Port Range	Source	
TCP	80	0.0.0.0/0	<a href="#">Remove</a>
TCP	443	0.0.0.0/0	<a href="#">Remove</a>

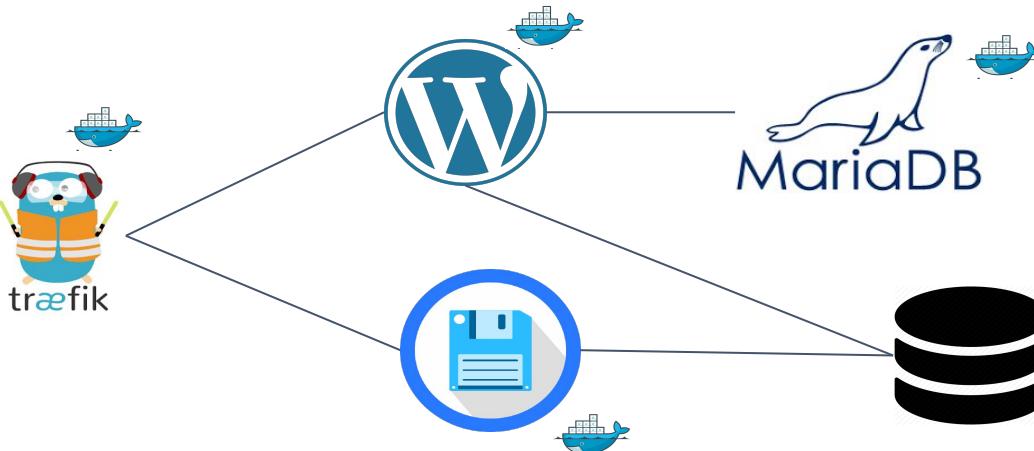
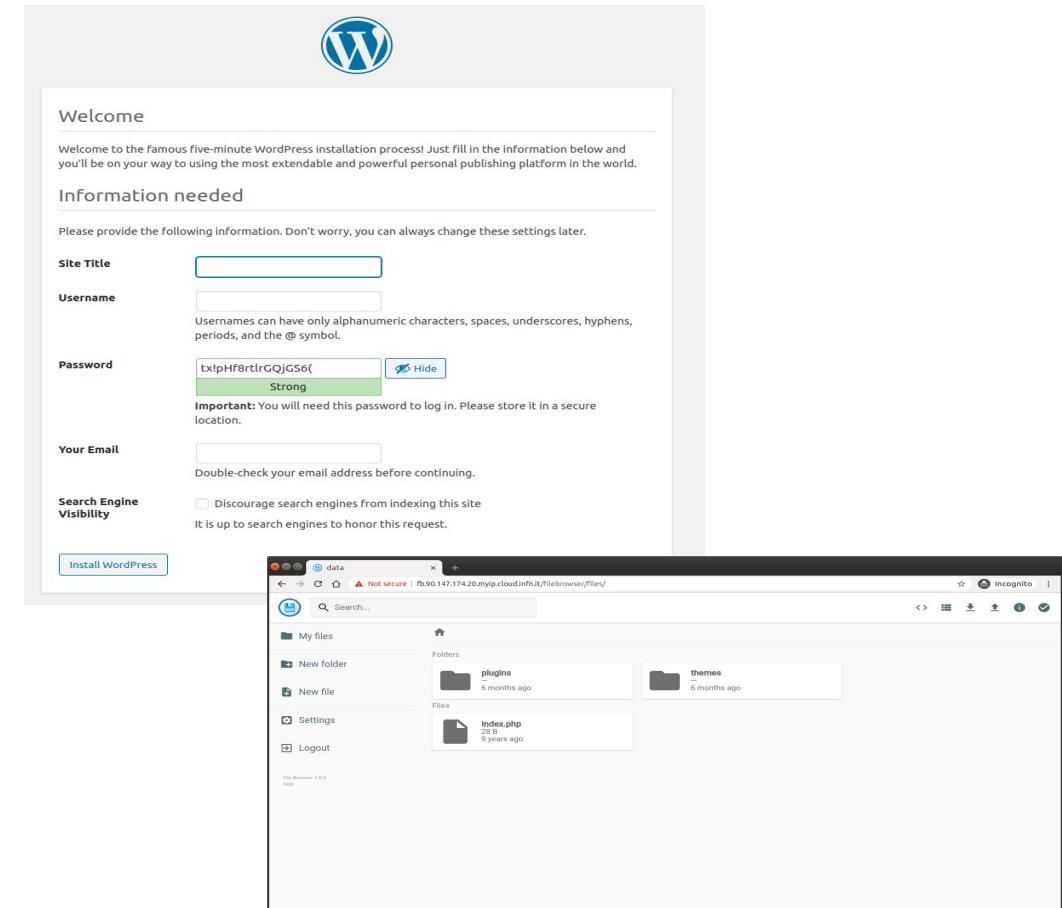
[Add rule](#)

Ports to open on the machine

# Docker compose example

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

Author: Stefano Stalio (LNGS)

The top screenshot shows the WordPress 'Welcome' page with fields for Site Title, Username, Password, Your Email, and Search Engine Visibility. The password field is highlighted as 'Strong'. The bottom screenshot shows a file browser interface with a tree view of 'My files' containing 'New folder', 'New file', 'Settings', and 'Logout'. Folders 'plugins' and 'themes' are shown under 'Folders', and a file 'Index.php' is listed under 'Files'.



# 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=web"  
      -  
      "--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.${HOST_PUBLIC_IP}.myip.cloud.infn.it  
    expose:  
      - 80  
  labels:  
    - "traefik.enable=true"  
    - "traefik.http.middlewares.app-redirect-ssl.redirectscheme.scheme=https"  
    - "traefik.http.routers.app-nossal.middlewares=app-redirect-ssl"  
    -  
    "traefik.http.routers.app-nossal.rule=Host(`wp.${HOST_PUBLIC_IP}.myip.cloud.infn.it`)"  
    - "traefik.http.routers.app-nossal.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

The screenshot shows a documentation page with the following structure:

- Header:** INFN Cloud logo, latest version, search bar.
- Table of Contents:** Getting Started, How To: Create VM with ssh access, How To: Configure the backup on your deployment, How To: Deploy Sync&Share aaS, How To: Deploy a Kubernetes cluster, How To: Deploy an Apache Mesos cluster, How To: Deploy a Spark cluster + Jupyter notebook, How To: Deploy Elasticsearch & Kibana, How To: Deploy RStudio Server, How To: Instantiate docker containers using custom docker-compose files (selected), Prerequisites, How to deploy and access docker-compose, How To: Instantiate docker containers using docker run, How To: Access cloud storage from a scientific environment.
- Page Content:**
  - Section:** How To: Instantiate docker containers using custom docker-compose files
  - Table of Contents:** How To: Instantiate docker containers using custom docker-compose files (selected), Prerequisites, How to deploy and access docker-compose, Step 1 - Connecting and authenticating to the INFN-CLOUD dashboard, Step 2 - Select and Configure the docker-compose deployment, Step 3 - Submitting the Docker-compose deployment, Step 4 - Access your application.
  - Text:** Make sure you are registered to the IAM system for INFN-CLOUD <https://iam.cloud.infn.it/login>. Only registered users can login into the INFN-CLOUD dashboard <https://paas.cloud.infn.it/login>.
  - Text:** Access to the INFN-CLOUD dashboard enables users to instantiate the "docker compose" deployment.
  - Section:** How to deploy and access docker-compose
  - Text:** Docker-compose allows you to instantiate fully functional production level services by using a set of coordinated dockers.
  - Text:** The public IP Address of the VM hosting the docker containers is available to the docker-compose file as an environment variable: HOST\_PUBLIC\_IP
  - Section:** Step 1 - Connecting and authenticating to the INFN-CLOUD dashboard
  - Text:** Connect to the INFN-CLOUD dashboard (<https://paas.cloud.infn.it/>).
  - Text:** You can authenticate with the credentials used for the IAM account (<https://iam.cloud.infn.it/login>) in order to access the dashboard.
- Footer:** Read the Docs, v: latest, INFN logo.

[https://guides.cloud.infn.it/docs/users-guides/en/latest/users\\_guides/howto7.html](https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto7.html)



# Elasticsearch + Kibana use-case

Deploy a virtual machine pre-configured with the Elasticsearch search and analytics engine and with Kibana for simple visualization of data with charts and graphs in Elasticsearch

# Service configuration and implementation

Elasticsearch and Kibana (version 8.4.1)

Description: Deploy a virtual machine pre-configured with the Elasticsearch search and analytics engine and with Kibana for simple visualization of data with charts and graphs in Elasticsearch

Deployment description  
ek

Configuration Advanced

contact\_email  
antonacci@infn.it

Insert your Email for receiving notifications

elastic\_password  
.....

Password for user elastic

kibana\_password  
.....

Password for user kibana\_system (internal user)

volume\_size  
10 GB

Size of the volume to be used to store the data

mountpoint  
/data

Path to mount the data volume

flavor  
large: 4 vCPUs, 8 GB RAM

Number of vCPUs and memory size of the Virtual Machine

**https://elastic.<IP>.myip.cloud.infn.it**

11ed351e-3748-9cba-b185-0242a79ac9f5

Description: ek

Overview Input values Output values

node\_ip: 192.135.24.13  
kibana\_username: elastic  
elasticsearch\_endpoint: https://elastic.192.135.24.13.myip.cloud.infn.it  
kibana\_endpoint: https://kibana.192.135.24.13.myip.cloud.infn.it  
ssh\_account: antonacci

Traefik

Elasticsearch

Kibana

**https://kibana.<IP>.myip.cloud.infn.it**

Traefik terminates the SSL connections: it is configured to use an ACME provider (Let's Encrypt) for automatic certificate generation.



# How to su guides.cloud.infn.it

Getting Started

- How To: Create VM with ssh access (sys-admin nomination required)
- How To: Deploy Sync&Share aaS (sys-admin nomination required)
- How To: Associate a FQDN to your VMs
- How To: Run JupyterHub on a single VM enabling Notebooks persistence (sys-admin nomination required)
- How To: Use the Notebooks as a Service solution
- How To: Request to open ports on deployed VMs
- How To: Deploy Working Station for CYGNO Experiment (sys-admin nomination required)
- How To: create a customized docker image for services
- How To: Use the INFN Cloud object storage service
- How To: Deploy a Kubernetes cluster (sys-admin nomination required)
- How To: Deploy an Apache Mesos cluster (sys-admin nomination required)
- How To: Deploy a Spark cluster + Jupyter notebook (sys-admin nomination required)
- How To: Deploy Elasticsearch & Kibana (sys-admin nomination required)

Prerequisites

- User responsibilities
- Elasticsearch & Kibana configuration
- Deployment result

How To: Deploy RStudio Server (sys-admin nomination required)

How To: Instantiate docker containers using custom docker-compose files (sys-admin nomination required)

How To: Instantiate docker containers using docker run (sys-admin nomination required)

How To: Access cloud storage from a scientific environment

How To: Request the "nomination to be system administrator"

How To: Request the "nomination to be system administrator" (Italian version)

[Read the Docs](#) [v: latest](#)

Docs » How To: Deploy Elasticsearch & Kibana (sys-admin nomination required) [View page source](#)

## How To: Deploy Elasticsearch & Kibana (sys-admin nomination required)

### Table of Contents

- How To: Deploy Elasticsearch & Kibana (sys-admin nomination required)
  - Prerequisites
  - User responsibilities
  - Elasticsearch & Kibana configuration
    - Basic configuration
    - Advanced configuration
  - Deployment result

### Prerequisites

The user has to be registered in the IAM system for INFN-Cloud <https://iam.cloud.infn.it/login>. Only registered users can login into the INFN-Cloud dashboard <https://my.cloud.infn.it/login>.

### User responsibilities

#### Important

The solution described in this guide consists on the deployment of Elasticsearch and Kibana on top of Virtual Machines instantiated on INFN-CLOUD infrastructure. The instantiation of a VM comes with the responsibility of maintaining it and all the services it hosts.

Please read the INFN Cloud AUP in order to understand the responsibilities you have in managing this service.

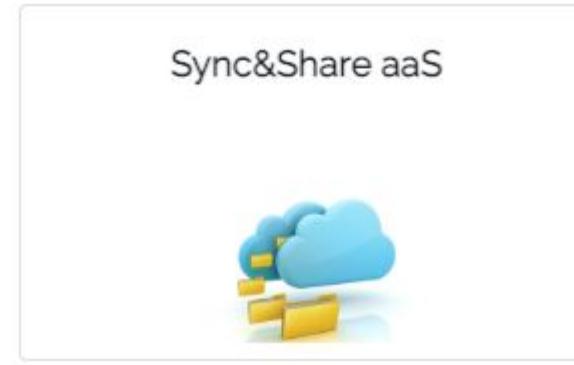
### Elasticsearch & Kibana configuration

#### Note

If you belong to multiple projects, aka multiple IAM-groups, after login into the dashboard, from the upper right corner, select the one to be used for the deployment you intend to perform. Not all solutions are available for all projects. The resources used for the deployment will be accounted to the respective project, and impact on their available **quota**. See figure below.

A screenshot of the INFN Cloud Dashboard. At the top, there's a navigation bar with links for 'Deployments', 'Advanced', 'External Links', and 'Users'. A dropdown menu shows 'infn-cloud-catchall' and 'Dolna Cristina Duma'. Below the navigation, there are two tabs: 'Virtual machine' and 'Docker-compose'. Under 'Virtual machine', there's a search bar and a table with several rows, one of which is highlighted in yellow. The table columns include 'Name', 'Status', 'Type', and 'Actions'. The 'Actions' column contains icons for edit, delete, and more. The URL in the browser address bar is 'https://my.cloud.infn.it/home'.

[https://guides.cloud.infn.it/docs/users-guides/en/latest/users\\_guides/howto5.html](https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto5.html)



# Sync&Share use-case

Easily share documentation and user data among a project or a scientific collaboration



# Sync&Share service

The INFN-Cloud Sync&Share aaS is currently based on the popular ***ownCloud*** storage solution.

INFN-Cloud users have full control over the configuration parameters of their Cloud Storage instance, as well as on third party accesses to the stored data.

Main features:

- **S3 based Object Storage backend** where data is replicated over two backbone data centers (CNAF, BARI)
- Authentication/Authorization based on **INFN-Cloud IAM** (via OIDC)
- **programmatic access** to user data via Rclone, including remote mount and folder sync
- embedded, automated DB and configuration **backup**
- embedded, pre-configured **monitoring system** with alert notifications

# Service configuration

Configuration   Advanced

docker\_storage\_size  
20 GB  
Size of the volume to be mounted in /var/lib/docker

contact\_email

Insert your Email for receiving notifications

owncloud\_admin\_username  
admin  
Username for ownCloud admin access

owncloud\_admin\_password  
 ⚒  
Password for ownCloud admin user

monitoring\_admin\_username  
admin  
Username for the admin user of the monitoring service

monitoring\_admin\_password  
 ⚒  
Password for the admin user of the monitoring service

backup\_passphrase  
 ⚒  
Password for backup

iam\_url  
https://iam.cloud.infn.it  
IAM url

iam\_authorized\_group  
  
IAM group authorized to access the service

flavor  
--Select--  
Number of vCPUs and memory size of the Virtual Machine

**Submit**   **Cancel**

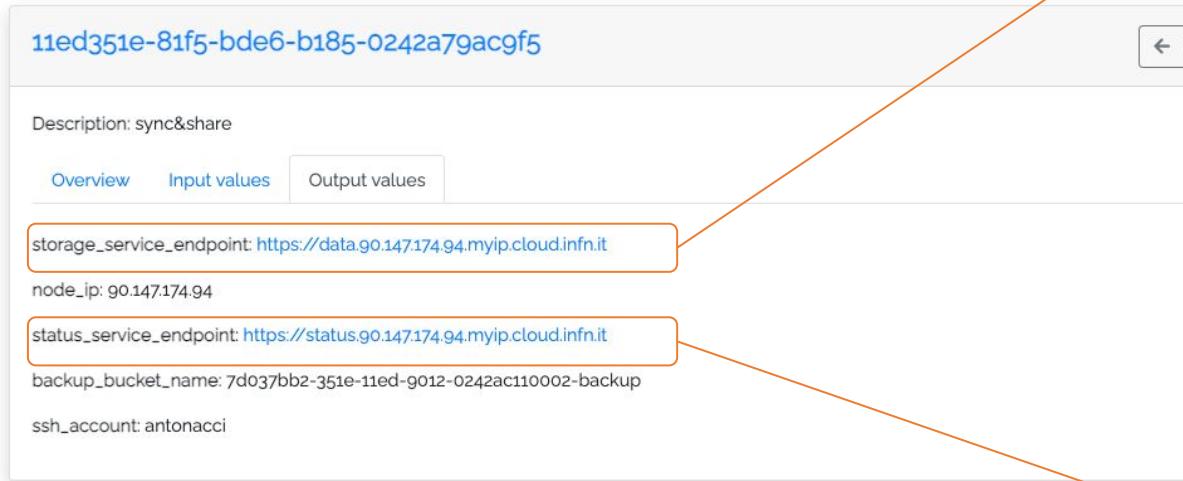
*Credentials for creating the owncloud admin user*

*Credentials for accessing the nagios dashboard*

*AuthN/AuthZ based on IAM*

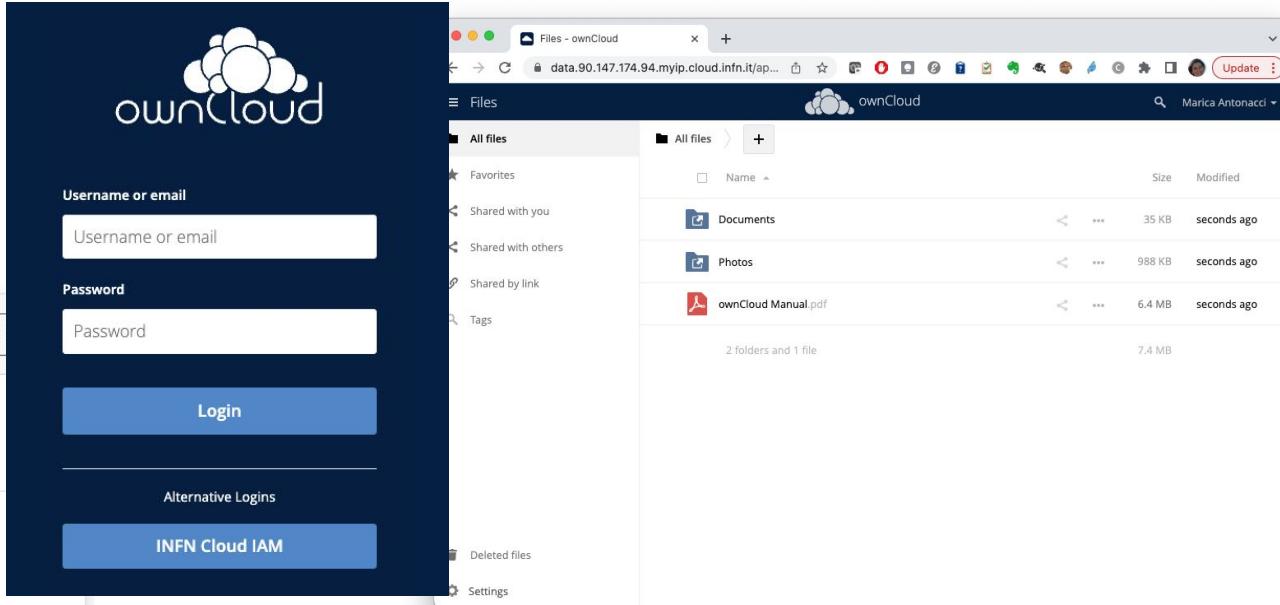
# Services are accessed through the reverse proxy based on Traefik

**https://data.<IP>.myip.cloud.infn.it**

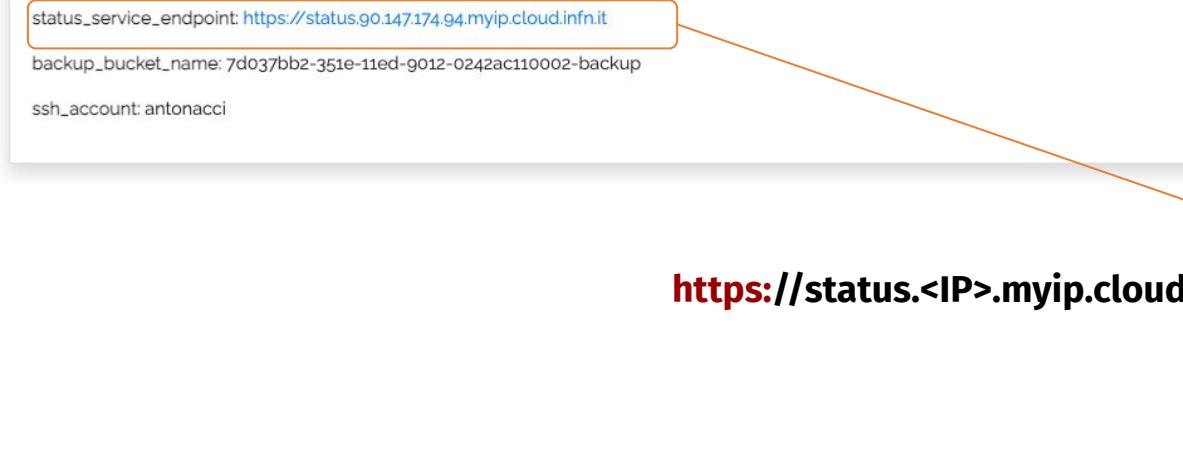


The screenshot shows a configuration page for a service endpoint. The URL is <https://data.11ed351e-81f5-bde6-b185-0242a79ac9f5>. The configuration includes:

- Description: sync&share
- Overview, Input values, Output values tabs
- storage\_service\_endpoint: <https://data.90.147.174.94.myip.cloud.infn.it> (highlighted)
- node\_ip: 90.147.174.94
- status\_service\_endpoint: <https://status.90.147.174.94.myip.cloud.infn.it> (highlighted)
- backup\_bucket\_name: 7d037bb2-351e-11ed-9012-0242ac110002-backup
- ssh\_account: antonacci

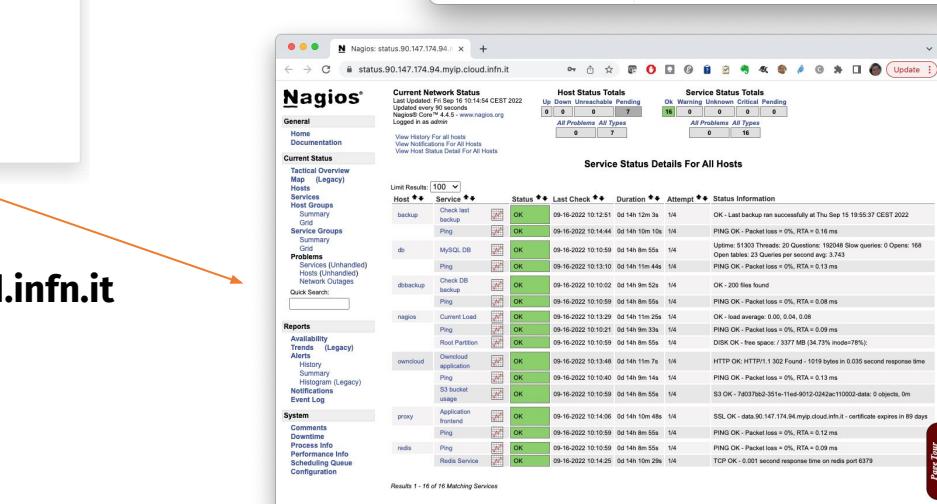


**https://status.<IP>.myip.cloud.infn.it**



The screenshot shows the Nagios monitoring interface. It displays the current network status, host status totals, and service status details for all hosts. The service status table lists various services across different hosts, including MySQL, PostgreSQL, and Redis, with their status, last check time, duration, and attempt information.

Service	Status	Last Check	Duration	Attempts	Information
MySQL, DB	OK	09-16-2022 10:50:59	0d 14h 1m 55s	1/4	PING OK - Packet loss = 0%, RTA = 0.13 ms
PostgreSQL, DB	OK	09-16-2022 10:50:59	0d 14h 1m 55s	1/4	PING OK - Packet loss = 0%, RTA = 0.13 ms
Redis Service	OK	09-16-2022 10:14:25	0d 14h 10m 29s	1/4	TCP OK - 0.001 second response time on redis port 6379





# How to su guides.cloud.infn.it

How To: Deploy Sync&Share aaS (sys-admin nomination required)

## Prerequisites

Sync&Share aaS Configuration

Deployment outputs

Access your services

Manage your service

Sync&Share aaS Backup

Sync&Share aaS Monitoring

How to access your data via cli

Troubleshooting

How To: Associate a FQDN to your VMs

How To: Run JupyterHub on a single VM enabling Notebooks persistence (sys-admin nomination required)

How To: Use the Notebooks as a Service solution

How To: Request to open ports on deployed VMs

How To: Deploy Working Station for CYGNO Experiment (sys-admin nomination required)

How To: create a customized docker image for services

How To: Use the INFN Cloud object storage service

How To: Deploy a Kubernetes cluster (sys-admin nomination required)

How To: Deploy an Apache Mesos cluster (sys-admin nomination required)

How To: Deploy a Spark cluster + Jupyter notebook (sys-admin nomination required)

How To: Deploy Elasticsearch & Kibana (sys-admin nomination required)

How To: Deploy RStudio Server (sys-admin nomination required)

How To: Instantiate docker containers using custom docker-compose files (sys-admin nomination required)

How To: Instantiate docker containers using docker run (sys-admin nomination required)

How To: Access cloud storage from a scientific environment

How To: Request the "nomination to be system administrator"

Read the Docs

v: latest ▾

Docs » How To: Deploy Sync&Share aaS (sys-admin nomination required)

[View page source](#)

## How To: Deploy Sync&Share aaS (sys-admin nomination required)

Sync&Share aaS is offered to INFN Cloud users as a means to easily share documentation and user data among a project or a scientific collaboration. Different tools should be used for scientific data archival and distribution.

### Table of Contents

- How To: Deploy Sync&Share aaS (sys-admin nomination required)
  - Prerequisites
  - Sync&Share aaS Configuration
    - Basic configuration tab
    - Advanced configuration tab
  - Deployment outputs
  - Access your services
  - Manage your service
    - Integration with IAM
  - Sync&Share aaS Backup
  - Sync&Share aaS Monitoring
  - How to access your data via cli
  - Troubleshooting
    - Authorization failure

### Prerequisites

The user has to be registered in the IAM system for INFN-CLOUD <https://iam.cloud.infn.it/login>. Only registered users can login into the INFN-CLOUD dashboard <https://my.cloud.infn.it>.

At the time being **additional authorizations are needed to submit a Sync&Share deployment on INFN Cloud**. Please contact the [support team](#) in order to be granted access to the INFN-Cloud Object storage backend **before** launching your application.

#### Important

- The solution described in this guide consists on deployment of a Sync&Share solution on top of a Virtual Machine instantiated on INFN-CLOUD infrastructure. The instantiation of a VM comes with the responsibility of **maintaining it and all the services it hosts**.  
Please read the INFN Cloud AUP in order to understand the responsibilities you have in managing this service.
- Production level Sync&Share aaS instances **should not be deployed within the infn-catch-all project** as the archived data are potentially accessible by extraneous users. If you need to deploy a Sync & Share service for production use make sure to do it within your experiment project or to ask for a dedicated one.

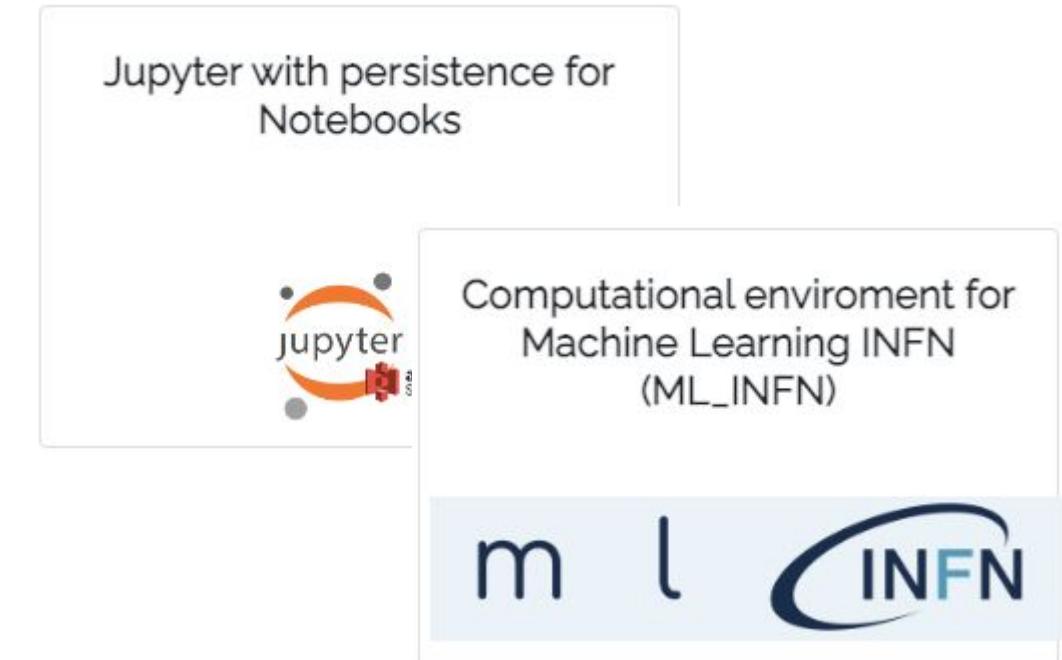
[https://guides.cloud.infn.it/docs/users-guides/en/latest/users\\_guides/howto11.html](https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto11.html)



# 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, ...

Computational environment for Machine Learning INFN (ML\_INFN)

Description: Run a single VM with exposing both ssh access and multiuser JupyterHub interface, integrating the ML-INFN environment

Deployment description

jupyter\_images

dodasts/mlinfn-base:v1 dodasts/mlinfn-conda-base:v2

Default image for jupyter server

cvmfs\_repos

cms.cern.ch sft.cern.ch atlas.cern.ch

CMFS repositories to mount

ports

Add rule

Ports to open on the VM

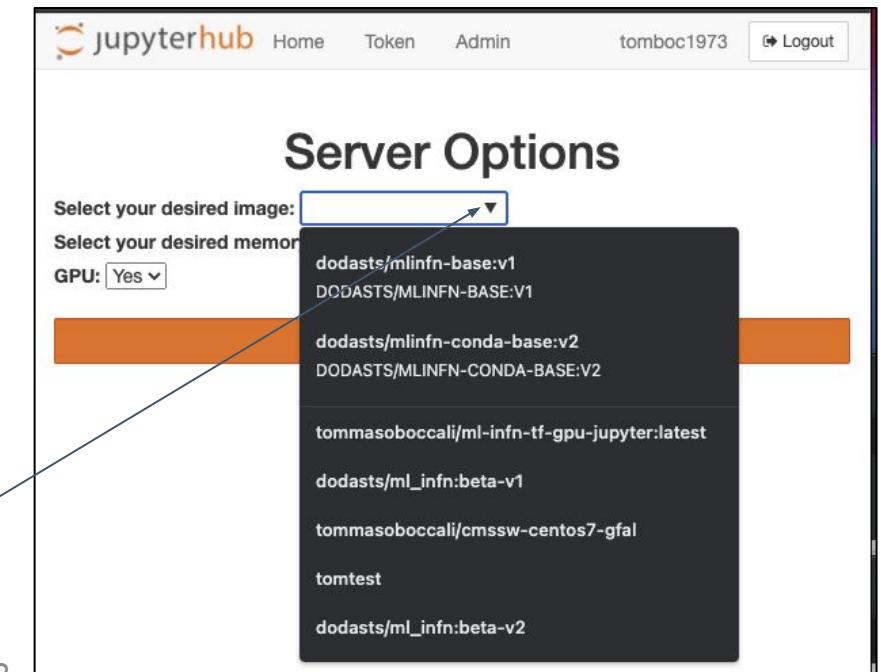
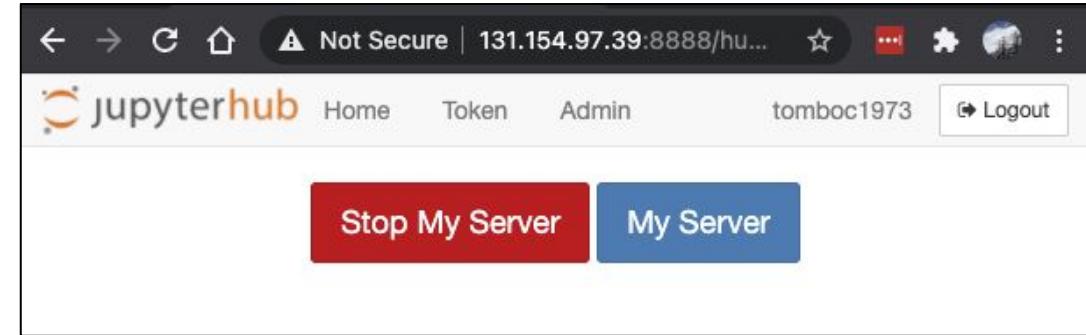
flavor

--Select--

- 16 VCPUs, 128 GB RAM, 512 GB disk, 1 GPU
- 16 VCPUs, 128 GB RAM, 1 TB disk, 1 GPU
- 8 VCPUs, 64 GB RAM, 512 GB disk, 2 GPU
- 8 VCPUs, 64 GB RAM, 1 TB disk, 2 GPU
- 16 VCPUs, 128 GB RAM, 512 GB disk, 2 GPU
- 16 VCPUs, 128 GB RAM, 1 TB disk, 2 GPU

# What is inside the VM?

- A **jupyterhub** runs in the VM, and allows **authorized users** to create their running instance through a container (started from an image hosted either locally, or on a registry like 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 via ssh



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->3000
6bead4f067ee        prom/prometheus:latest    "/bin/prometheus --c..."  11 days ago       Up 11 days       0.0.0.0:9090->9090
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->8099
18cc7311bf14        mircot/jupyterlab_collaborative:ml_base  "jupyter lab --ip=0..."  11 days ago       Up 11 days       0.0.0.0:8889->8889
e0f479af4a86        jupyterhub_backup_service   "cron -f"              11 days ago       Up 11 days       0.0.0.0:8001->8001
db642fee83e3        jupyterhub/configurable-http-proxy  "/srv/configurable-h..." 11 days ago       Up 11 days       0.0.0.0:8001->8001
root@vnode-0:/home/spiga#
```

# Access as “User”

The screenshot shows the JupyterHub dashboard. At the top, there are links for "Logout" and "Control Panel". Below that, tabs for "Files", "Running", and "IPython Clusters" are visible. A message says "Select items to perform actions on them." On the left, there's a sidebar with checkboxes for "0", "cvmfs", "private", and "shared". A dropdown menu is open, showing options for "Notebook" (Python 3, ROOT C++) and "Other" (Text File, Folder, Terminal). Buttons for "Upload" and "New" are also present.

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

**Access granted via notebooks and via terminal**

The terminal window shows the following output:

```
# bash
=====
WARNING: You are running this container as root, which can cause new files in mounted volumes to be created as the root user on your host machine.

To avoid this, run the container by specifying your user's userid.

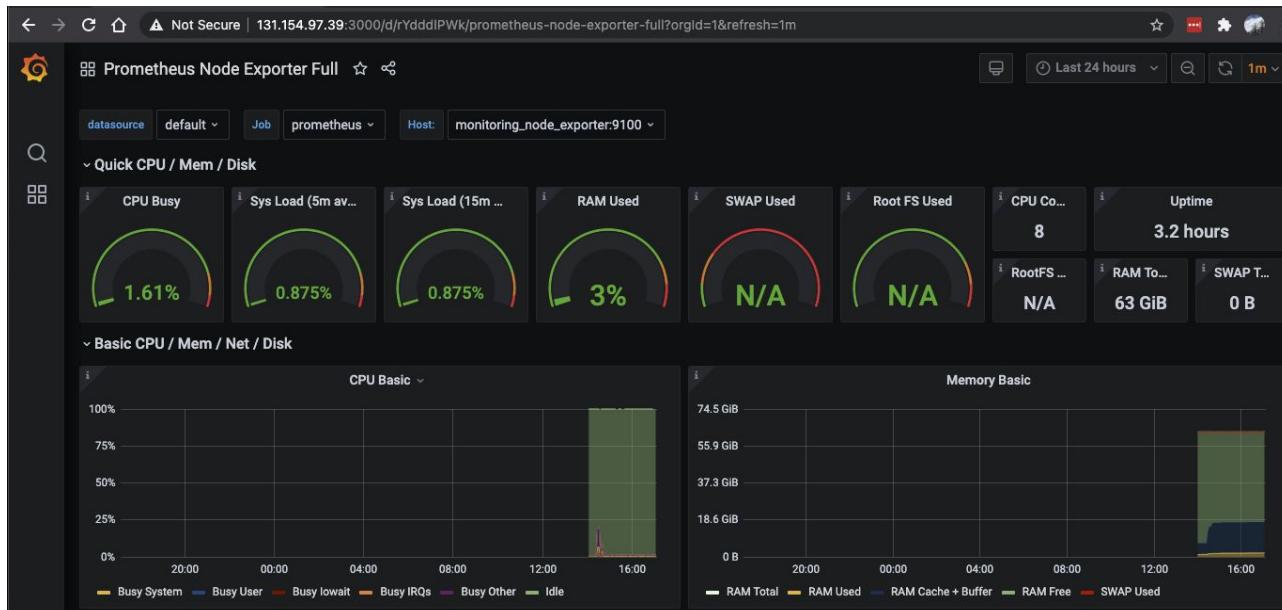
$ docker run -u $(id -u):$(id -g) args...
root@bd8f6fe3da4c:/workarea# nvidia-smi
Sun Mar 21 15:58:10 2021
+-----+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2 |
+-----+
| GPU  Name      Persistence-M | Bus-Id     Disp.A  Volatile Uncorr. ECC | | | | | | |
| Fan  Temp     Perf  Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M. |
|          |          |          |          |      |          |          | MIG M. |
+-----+
| 0  Tesla T4      On           00000000:00:05.0 Off   0MiB / 15109MiB | 0%     Default N/A |
| N/A  34C     P8    9W / 70W |
+-----+
| 1  Tesla T4      On           00000000:00:06.0 Off   0MiB / 15109MiB | 0%     Default N/A |
| N/A  35C     P8    9W / 70W |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU  GI  CI   PID  Type  Process name        Usage  |
| ID  ID
+-----+
| No running processes found
+-----+
root@bd8f6fe3da4c:/workarea#
```

**Root access, 2 GPUs available**



# Monitoring etc

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



User				Admin		Last Activity		Running (3)	
Add Users				Start All	Stop All	Shutdown Hub			
cduma	admin	3 minutes ago	<button>stop server</button>	<button>edit user</button>	<button>delete user</button>				
spiga	admin	2 hours ago	<button>stop server</button>	<button>edit user</button>	<button>delete user</button>				
tomboc1973	admin	a few seconds ago	<button>stop server</button>	<button>edit user</button>					

Displaying users 1 - 3 of 3

JupyterHub 1.3.0 20210321125835

The screenshot shows the JupyterHub 1.3.0 administration interface. It includes a header with navigation links and a search bar. Below the header is a table of users with columns for Name, Admin status, Last Activity, and actions (stop server, edit user, delete user). A message at the bottom indicates "Displaying users 1 - 3 of 3". The main content area has sections for "General", "Docker and system monitoring", "NVIDIA DCGM Exporter Dashboard", and "Prometheus Node Exporter Full". A red "linux" button is visible in the bottom right corner.



# Kubernetes cluster use-case

How to deploy a complete k8s cluster on INFN Cloud

# Configure your cluster

Kubernetes cluster

Description: Deploy a single master Kubernetes 1.23.8 cluster

Deployment description

description

Configuration Advanced

admin\_token

Enter your password

password token for accessing k8s dashboard

number\_of\_nodes

1

number of K8s node VMs

ports

Add rule

Ports to open on the K8s master VM

master\_flavor

--Select--

Number of vCPUs and memory size of the k8s master VM

node\_flavor

--Select--

Number of vCPUs and memory size of each k8s node VM

Submit Cancel

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

11ed2e89-cfc7-d325-9808-02424e495bee

Description: kubernetes

Overview   Input values   Output values

k8s\_node\_ip: '172.30.143.162'  
grafana\_endpoint: <https://grafana.90.147.102.94.myip.cloud.infn.it>  
grafana\_username: admin  
k8s\_master\_ip: 90.147.102.94  
k8s\_endpoint: <https://dashboard.90.147.102.94.myip.cloud.infn.it>  
ssh\_account: antonacci

kubeconfig:

[Download](#) [Copy to clipboard](#)

The image displays three screenshots illustrating the access and monitoring of Kubernetes services:

- Kubeconfig Download:** A screenshot of a web interface showing a kubeconfig file. It includes fields for 'k8s\_node\_ip' (172.30.143.162), 'grafana\_endpoint' (<https://grafana.90.147.102.94.myip.cloud.infn.it>), 'grafana\_username' (admin), 'k8s\_master\_ip' (90.147.102.94), 'k8s\_endpoint' (<https://dashboard.90.147.102.94.myip.cloud.infn.it>), and 'ssh\_account' (antonacci). Below these fields are two buttons: 'Download' and 'Copy to clipboard'. An orange arrow points from this interface to the Grafana dashboard.
- Grafana Cluster Monitoring:** A screenshot of the Grafana interface titled 'General / Cluster Monitoring for Kubernetes'. It shows various metrics: 'Cluster memory usage' (44.3%, 3.40 GiB / 7.68 GiB), 'Cluster CPU usage (2m avg)' (7.73%), and 'Cluster filesystem usage' (26.50%, 138.95 GiB / 524.42 GiB). The Grafana URL is [https://grafana.192.135.24.151.myip.cloud.infn.it/d/JABGX\\_-mz/cluster-monitoring-10](https://grafana.192.135.24.151.myip.cloud.infn.it/d/JABGX_-mz/cluster-monitoring-10). An orange arrow points from the 'grafana\_endpoint' field in the first screenshot to this dashboard.
- Kubernetes Workloads:** A screenshot of the Kubernetes dashboard under the 'Workloads' tab. It displays four main categories: Daemon Sets (Running: 3), Deployments (Running: 12), Pods (Running: 22), and Replica Sets (Running: 12). Below these are detailed tables for each category. For example, the 'Deployments' table lists the 'grafana' deployment with 1 pod running, managed by Helm, and using the 'grafana/grafana:8.5.3' image. An orange arrow points from the 'k8s\_endpoint' field in the first screenshot to this dashboard.



# How to su guides.cloud.infn.it

Search docs

## TABLE OF CONTENTS

- Getting Started
- How To: Create VM with ssh access (sys-admin nomination required)
- How To: Deploy Sync&ShareaaS (sys-admin nomination required)
- How To: Associate a FQDN to your VMs
- How To: Run JupyterHub on a single VM enabling Notebooks persistence (sys-admin nomination required)
- How To: Use the Notebooks as a Service solution
- How To: Request to open ports on deployed VMs
- How To: Deploy Working Station for CYGNO Experiment (sys-admin nomination required)
- How To: Deploy a Kubernetes cluster (sys-admin nomination required)
  - Prerequisites
  - User responsibilities
  - Kubernetes cluster configuration
  - Deployment result
- How To: Deploy an Apache Mesos cluster (sys-admin nomination required)
- How To: Deploy a Spark cluster + Jupyter notebook (sys-admin nomination required)
- How To: Deploy Elasticsearch & Kibana (sys-admin nomination required)
- How To: Deploy RStudio Server (sys-admin nomination required)
- How To: Instantiate docker containers using custom docker-compose files (sys-admin nomination required)
- How To: Instantiate docker containers using docker run (sys-admin nomination required)
- How To: Access cloud storage from a scientific environment
- How To: Request the "nomination to be system administrator"
- How To: Request the "nomination to be system administrator" (italian version)

Read the Docs v: latest ▾

Docs » How To: Deploy a Kubernetes cluster (sys-admin nomination required) View page source

## How To: Deploy a Kubernetes cluster (sys-admin nomination required)

### Table of Contents

- How To: Deploy a Kubernetes cluster (sys-admin nomination required)
  - Prerequisites
  - User responsibilities
  - Kubernetes cluster configuration
    - Basic configuration
    - Advanced configuration
  - Deployment result
  - Troubleshooting

### Prerequisites

The user has to be registered in the IAM system for INFN-Cloud <https://iam.cloud.infn.it/login>. Only registered users can login into the INFN-Cloud dashboard <https://my.cloud.infn.it/login>.

### User responsibilities

#### Important

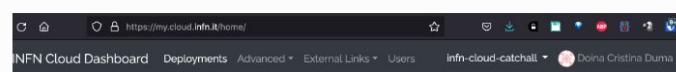
The solution described in this guide consists on the deployment of a Kubernetes cluster on top of Virtual Machines instantiated on INFN-CLOUD infrastructure. The instantiation of a VM comes with the responsibility of maintaining it and all the services it hosts. In particular, be careful when updating the operating system packages, as they could incorrectly modify the current version of the cluster (v1.19) and cause it to malfunction.

Please read the INFN Cloud AUP in order to understand the responsibilities you have in managing this service.

### Kubernetes cluster configuration

#### Note

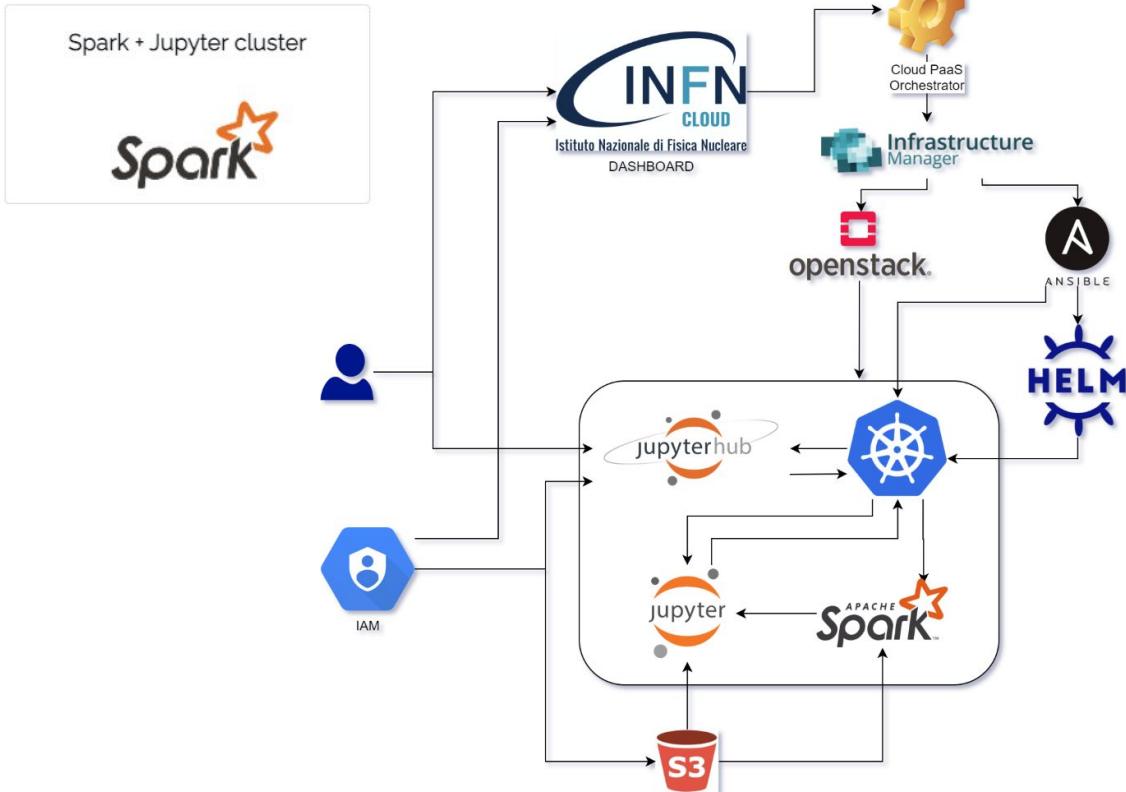
If you belong to multiple projects, aka multiple IAM-groups, after login into the dashboard, from the upper right corner, select the one to be used for the deployment you intend to perform. Not all solutions are available for all projects. The resources used for the deployment will be accounted to the respective project, and impact on their available quota. See figure below.

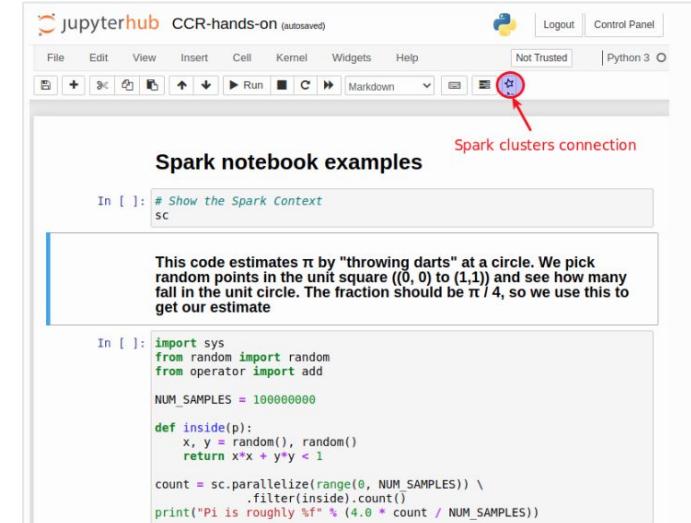


[https://guides.cloud.infn.it/docs/users-guides/en/latest/users\\_guides/howto2.html](https://guides.cloud.infn.it/docs/users-guides/en/latest/users_guides/howto2.html)

# Advanced k8s-based services

## Jupyter + Spark + K8s





Spark clusters connection

Figure 7: Jupyter notebook example.

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

Spark clusters connection

You are going to connect to:  
local

You can configure the following options.  
Environment variables can be used via {ENV\_VAR\_NAME}.

Add a new option

Bundled configurations  
These options will be overwritten by non-bundled options if specified

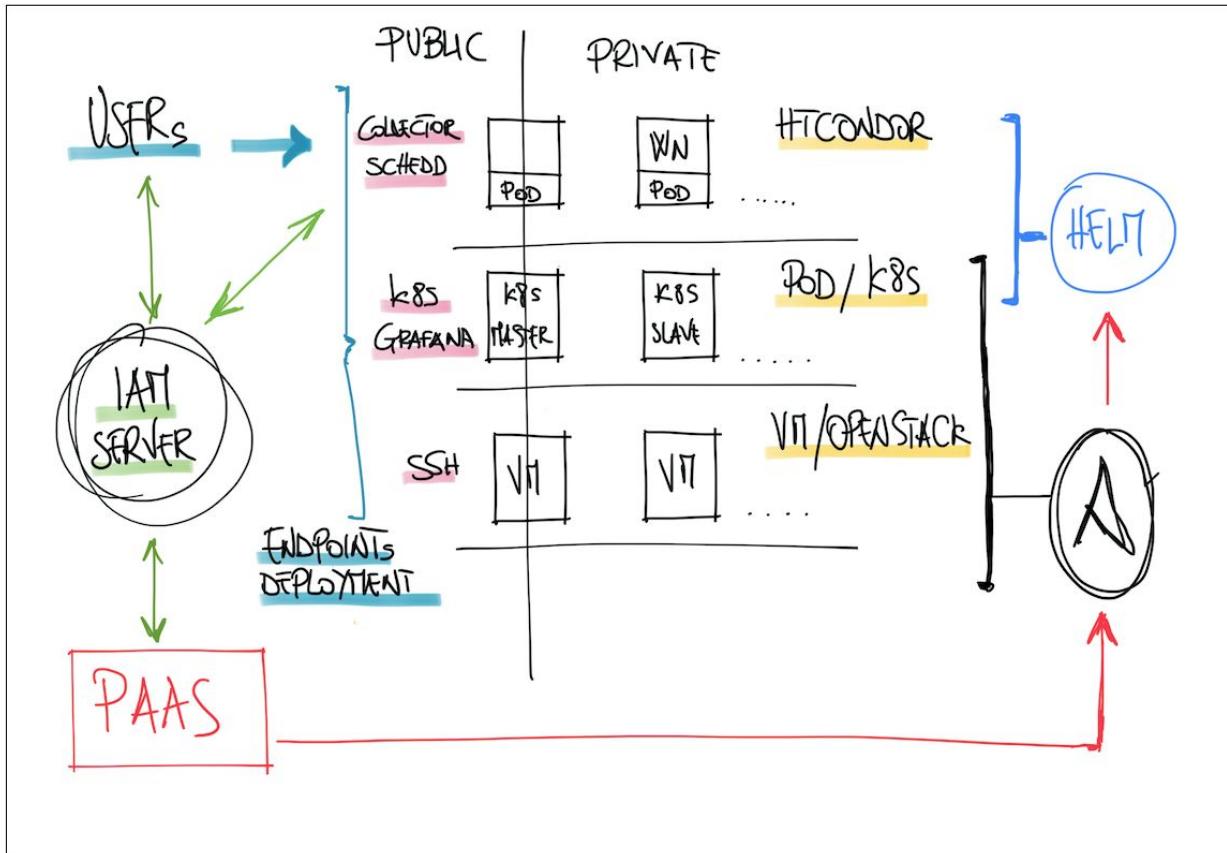
Selected configuration

- spark.executor.instances
  - 2

Connect

# Advanced k8s-based services (2)

## HTCondor + K8s



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**

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



# References

**Web site:**  
<https://www.cloud.infn.it>

A screenshot of the INFN Cloud website homepage. The header features the INFN Cloud logo and navigation links: Home, About us, Services, Resources, Documentation, News & Events, Training, and Contacts. Below the header is a dark banner with the text "Cloud Resources for research" and several cloud icons. At the bottom, a footer note states: "INFN is offering to its users a comprehensive and integrated set of Cloud services through its dedicated [INFN Cloud infrastructure](#)".

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

A screenshot of the "Welcome to the INFN Cloud Use Cases Documentation" page. The page has a blue header and a white content area. It includes a breadcrumb trail ("Docs &gt; Welcome to the INFN Cloud Use Cases Documentation"), a "View page source" link, and a "Table of Contents" section with links to "Getting Started", "How To: Create VM with ssh access", "How To: Configure the backup on your deployment", and "How To: Deploy Sync&amp;Share aaS".

**Support :**  
<https://servicedesk.cloud.infn.it> or [✉️ cloud-support@infn.it](mailto:cloud-support@infn.it)

A screenshot of the INFN CCR Help Center page. The header features the INFN Cloud logo and the text "INFN CCR Help Center" and "INFN Cloud". The main content area says "Welcome! You can raise a request to INFN Cloud from the options provided." Below this is a search bar with the placeholder "What do you need help with? Search". To the right, there are sections for "General", "Training", "Suggestions", and "Services trial questions". The "Services trial questions" section includes a note: "Want to see if our services can fit your workflows? Trying out our services and want more information? Select this and we'll be happy to answer your questions.".



Thank you  
for your attention!

Q&A

[www.cloud.infn.it](http://www.cloud.infn.it)

For general communications email us at [\*\*cloud@lists.infn.it\*\*](mailto:cloud@lists.infn.it)

To ask for support write to our mailing list [\*\*cloud-support@infn.it\*\*](mailto:cloud-support@infn.it), integrated with our [ServiceDesk](#)