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

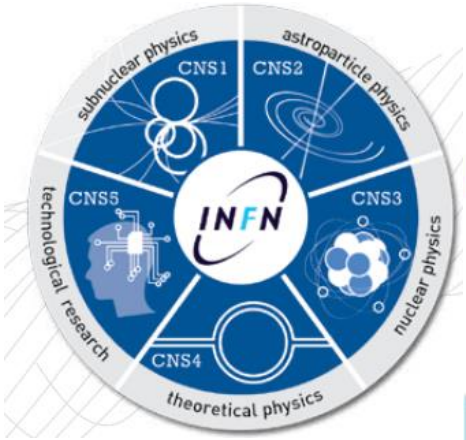
# Deploying the INDIGO PaaS Orchestration System Components on a Kubernetes-Based Infrastructure Using ArgoCD

L. Giommi – INFN CNAF

F. Sinisi, G. Savarese, A. Costantini, G. Donvito

International Symposium on Grids & Clouds (ISGC) 2026 | 15-20 March 2026

# INFN and its facilities




- INFN is the coordinating institution for **nuclear, particle, theoretical, and astroparticle physics** in Italy. It promotes, coordinates, and carries out scientific research as well as the **technological development** necessary for the activities in these sectors


- INFN manages and supports the **largest public computing infrastructure for scientific research** spread throughout the country

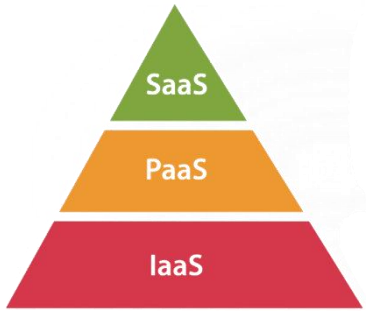
- INFN was one of main promoters of the GRID project to address LHC computing needs. Since then INFN has been participating to **WLCG** that includes more than 170 sites around the world, loosely organized in a tiered model.
  - In Italy, there are the Tier-1 at CNAF, Bologna and 9 Tier-2 centers

# The INFN Cloud ecosystem

INFN decided to implement a **national Cloud computing infrastructure** for research

- as a **federation** of existing distributed Cloud infrastructures
- as an "user-centric" infrastructure which makes 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, e.g. INDIGO DataCloud

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




e.g. Notebook as a Service

e.g. Kubernetes cluster

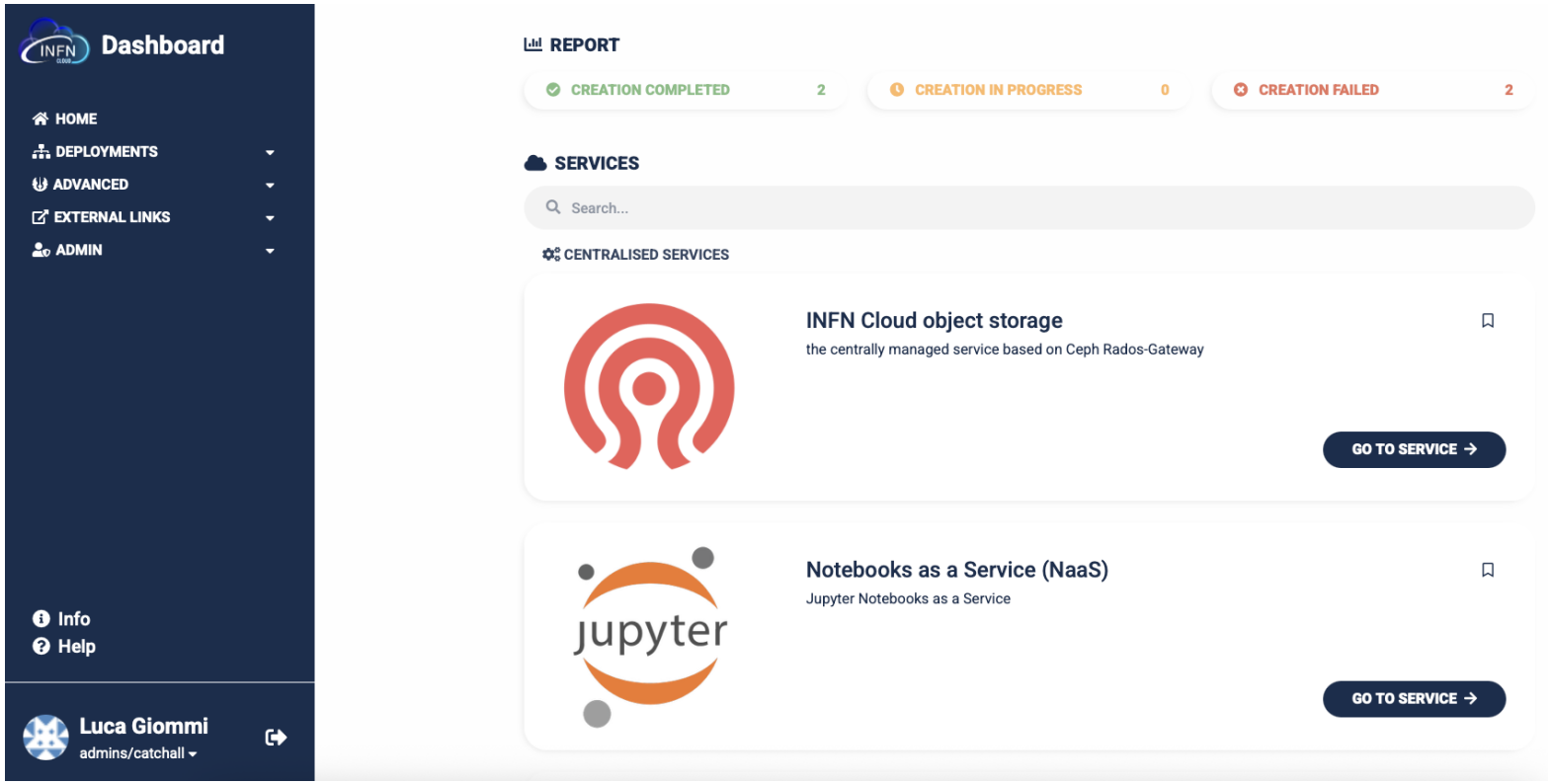
e.g. Virtual Machine


# The Infrastructure as Code paradigm

All PaaS services are defined using an **Infrastructure as Code** paradigm that aims to reduce manual processes and increase flexibility and portability across environments, via a combination of:

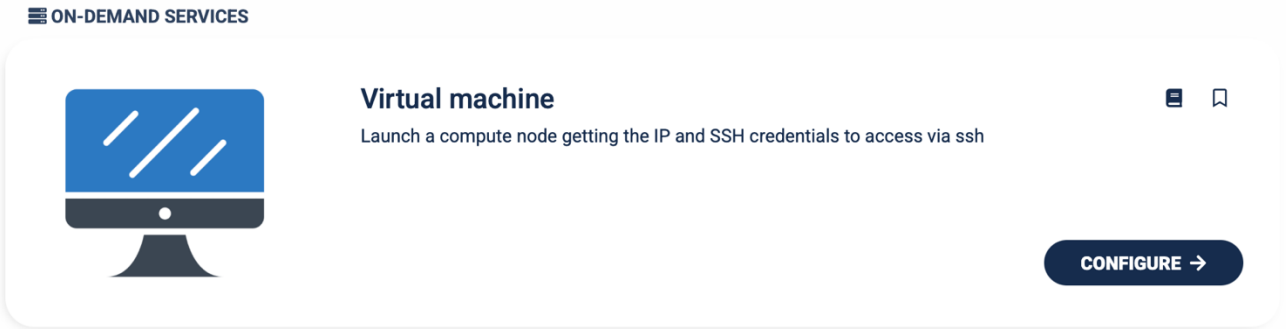
- **TOSCA** (**T**opology and **O**rchestration **S**pecification for **C**loud **A**pplications) templates, to model an application stack
- **Ansible** roles, to manage the automated configuration of virtual environments
- **Docker** containers, to encapsulate high-level application software and runtime
- **Helm** charts, to manage the deployment of an application in Kubernetes clusters

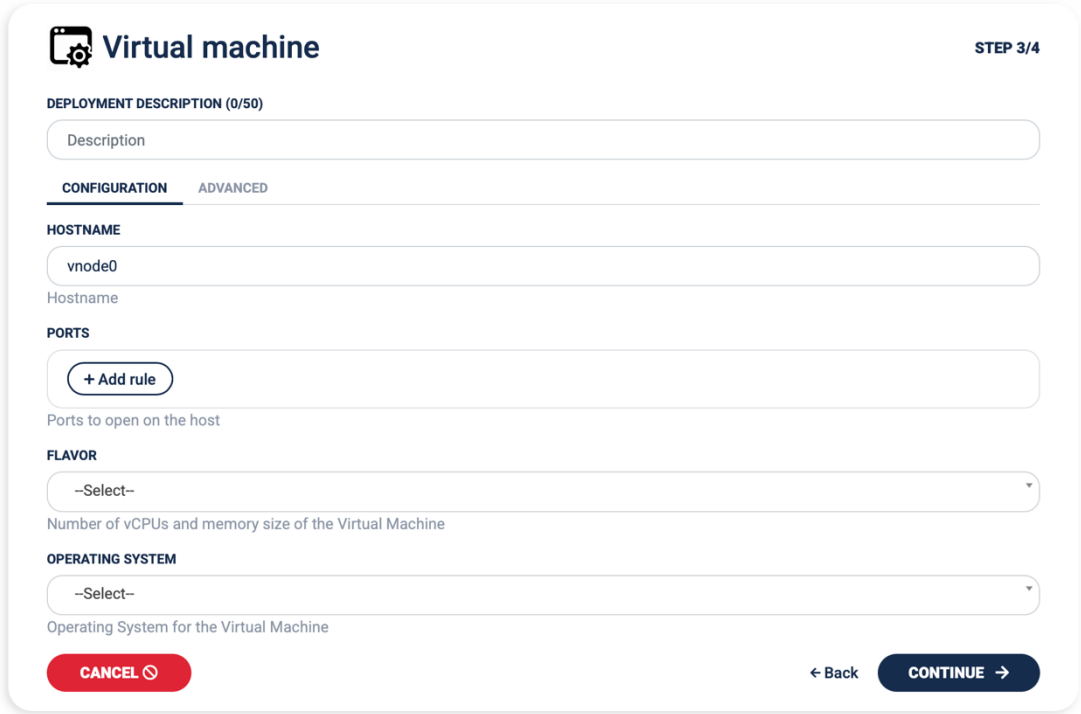
# The PaaS Orchestrator Dashboard



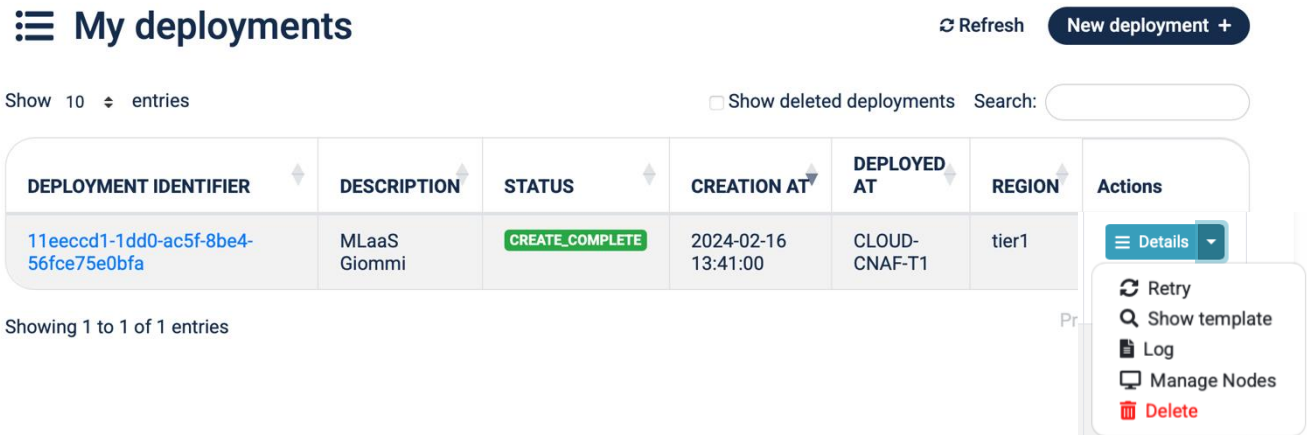

https://my.cloud.infn.it

# Deploy a PaaS service

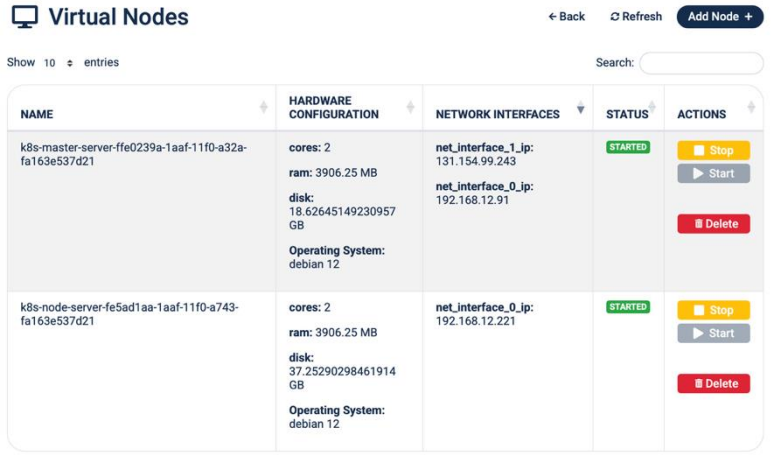


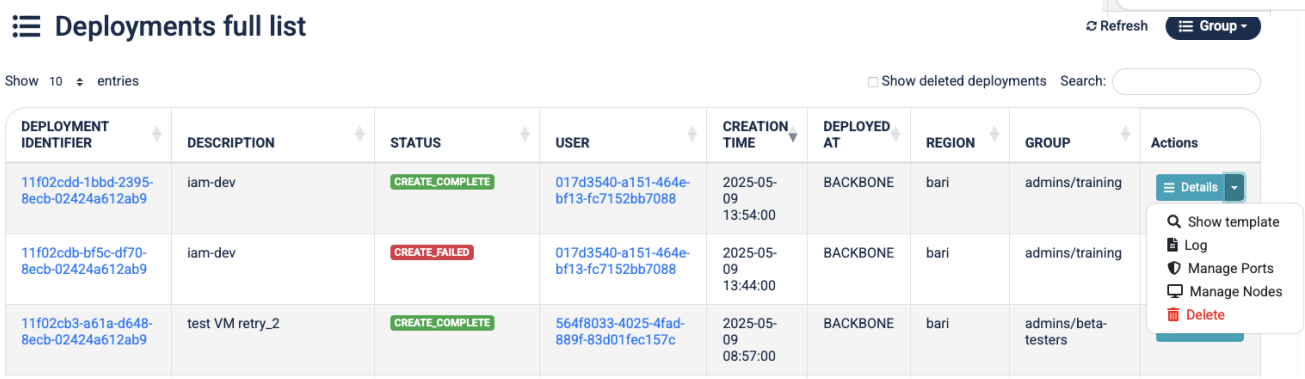



# Features of the PaaS Orchestrator dashboard

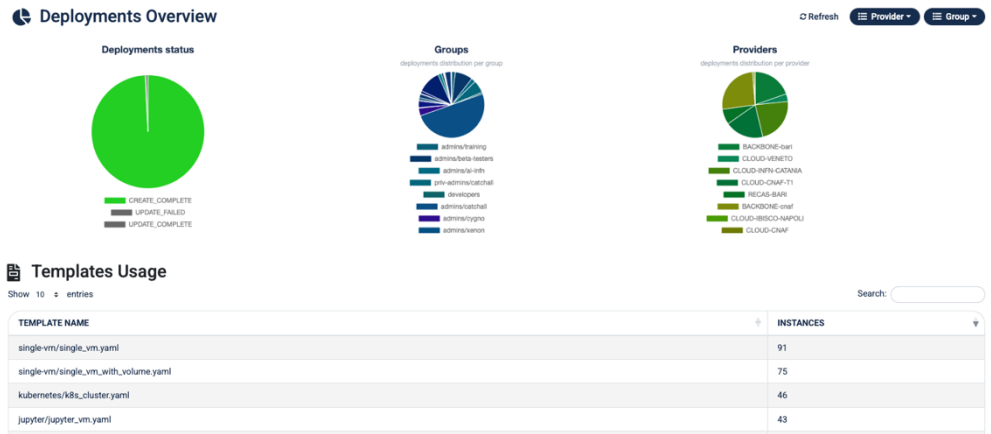



User view



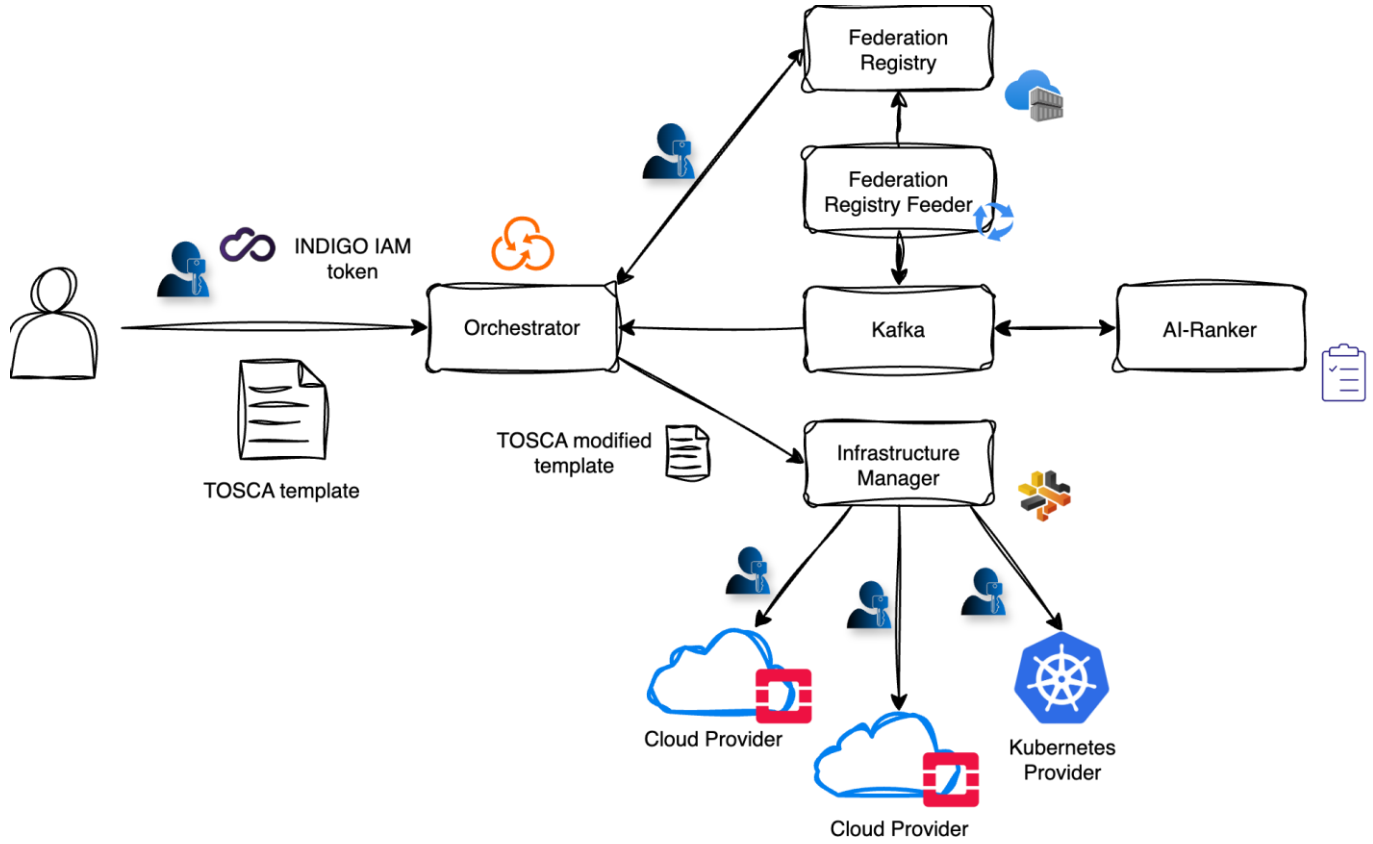




Admin view



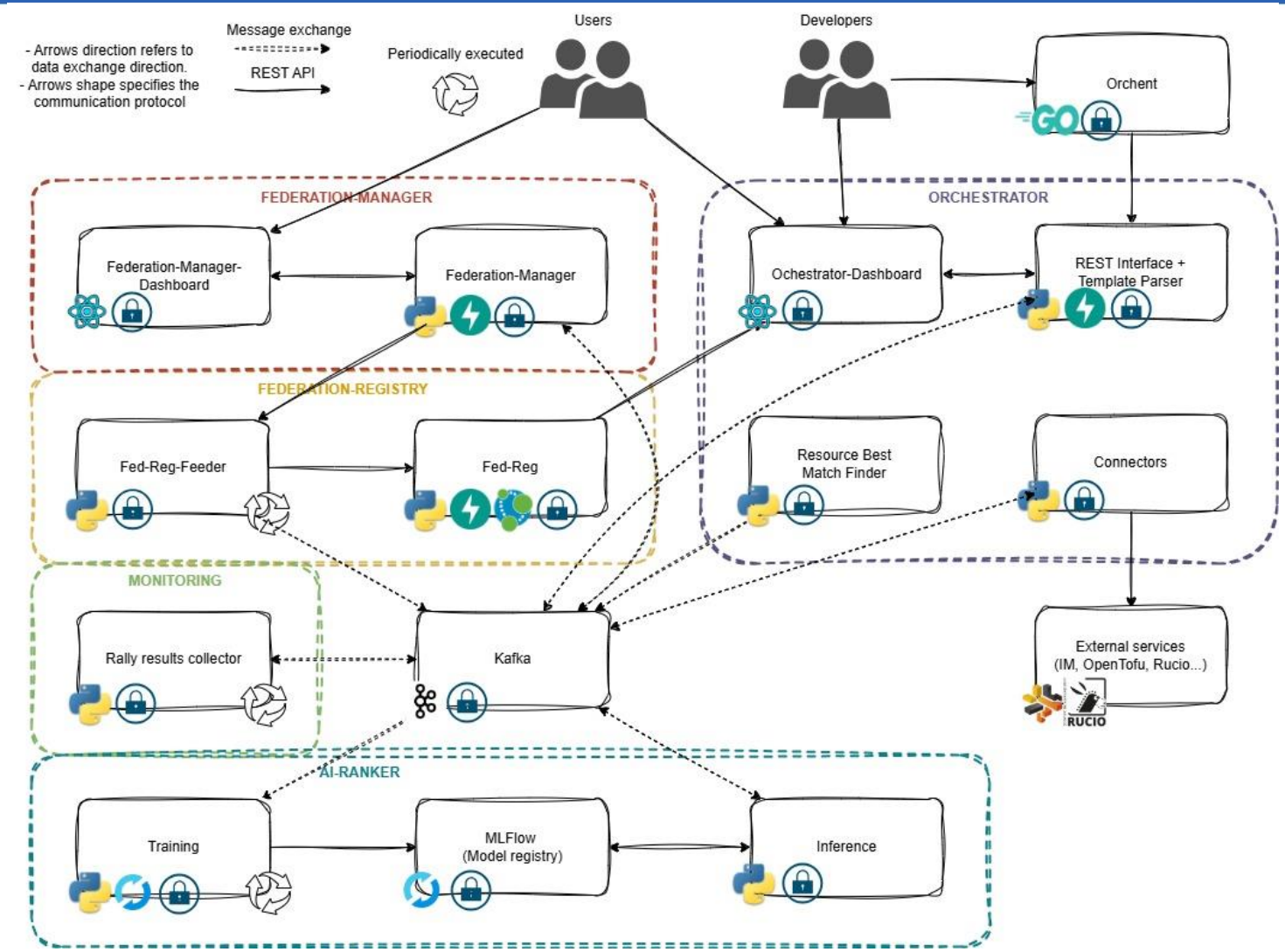
# The current INDIGO PaaS Orchestration system of INFN Cloud

- The federative middleware of INFN Cloud is based on the **INDIGO PaaS Orchestration system**, consisting of interconnected open-source microservices
- The Orchestrator receives high-level deployment requests in the form of TOSCA templates and coordinates the deployment process by using the **Infrastructure Manager (IM)** to interact with provider services and deploy complex, customized virtual infrastructures on the IaaS platforms offered by the federated providers
- A central activity was the introduction of the **Federation Registry, Feeder**, and **AI-Ranker** which replaced old components and were integrated with the entire PaaS Orchestration system

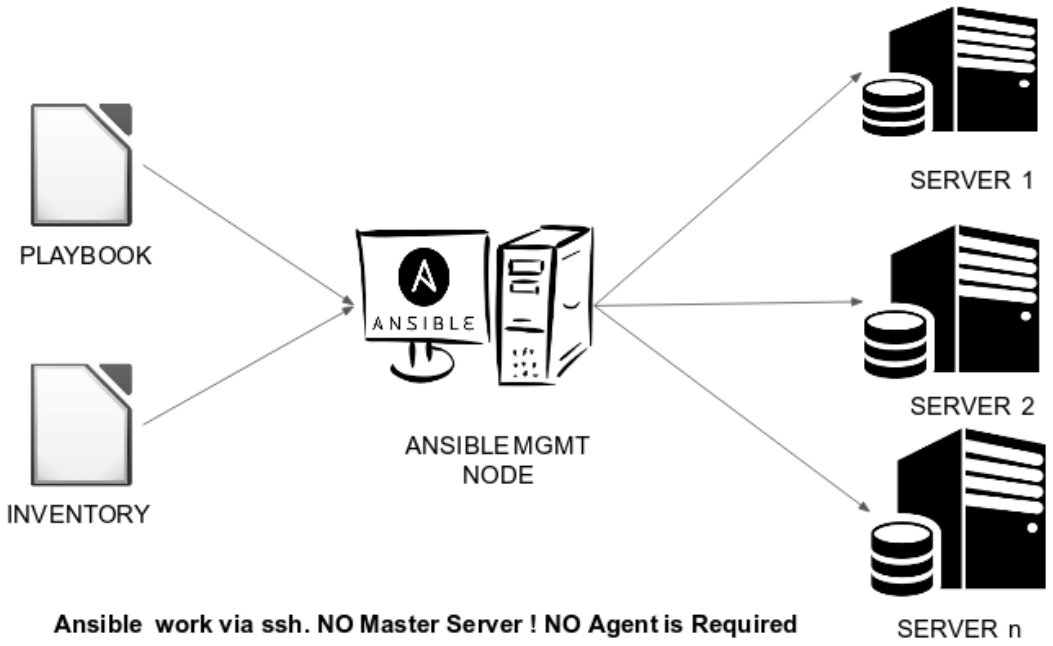


# The new PaaS architecture

- Rewriting of the Orchestrator from Java to Python
- Introduction of **Kafka** as streaming platform. The different services/components of the PaaS Orchestration system read/write data from/to queues
- Introduction of the **Federation Manager** to manage the entire federation process of new providers



# Current service deployment model (VM-based with Ansible)




Services of the PaaS Orchestration system are currently deployed on multiple VMs using Ansible playbook

**Cons**

- **Procedural, playbook-driven model**
  - No continuous state enforcement
  - Higher risk of configuration drift
- **Limited scalability**
  - VM-based management becomes complex at scale
  - Not aligned with Kubernetes / GitOps models

# Target Architecture: Kubernetes + ArgoCD (GitOps Model)

Applications are deployed on Kubernetes and managed through ArgoCD using a GitOps approach

### Declarative & GitOps-Driven

- Desired state defined in Git, Automatic reconciliation of cluster state, Built-in versioning and rollback

### Continuous Deployment & Drift Detection

- Automatic sync with repository, Immediate detection of configuration drift, Reduced manual intervention

### Scalability & Resilience

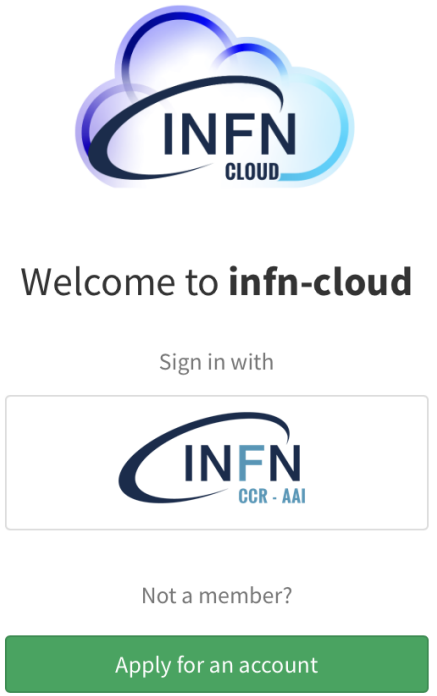
- Self-healing workloads, Horizontal scaling, High availability by design

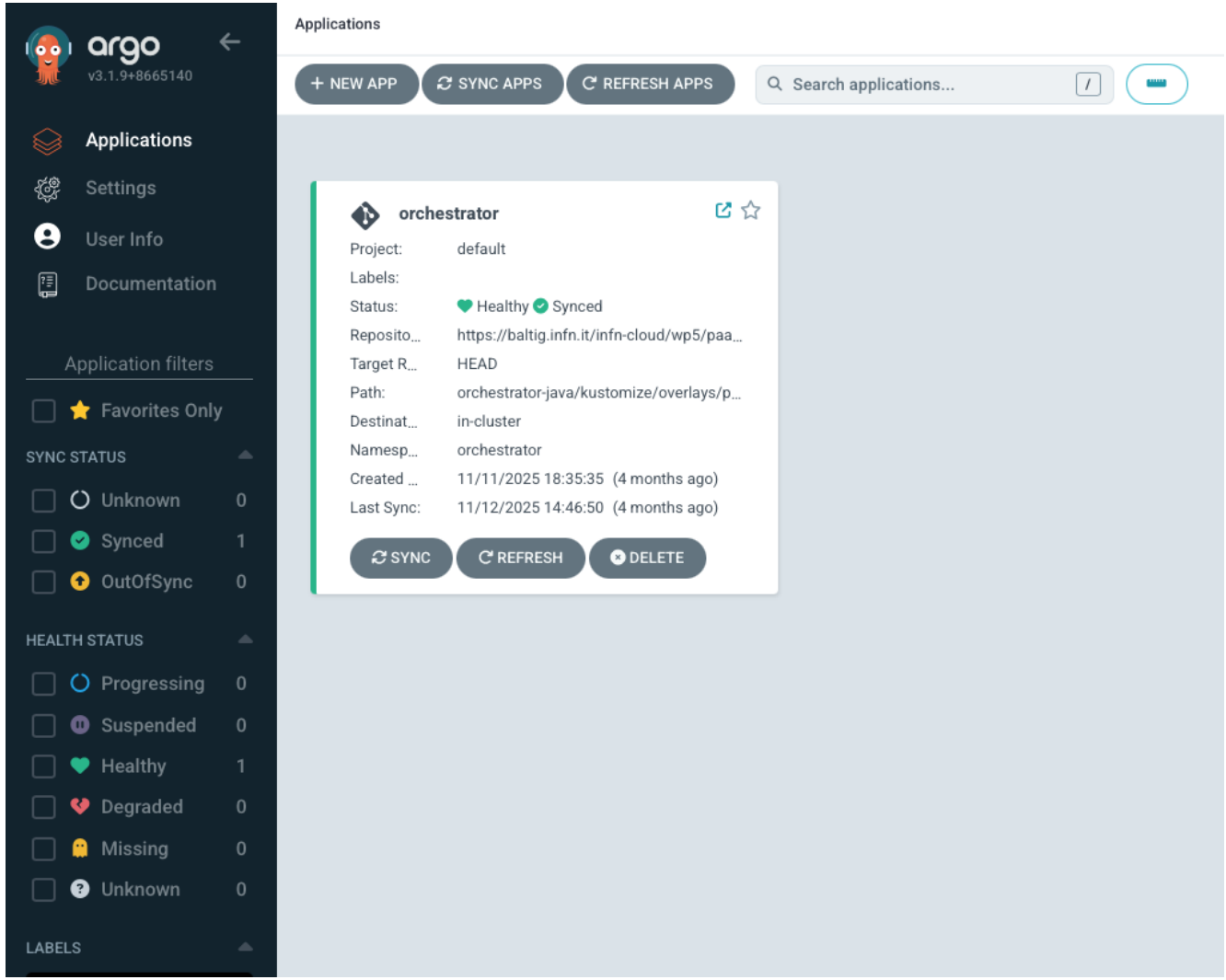
### Operational Simplicity at Scale

- Standardized deployment model, Environment consistency (dev / test / prod), Better alignment with modern cloud-native practices

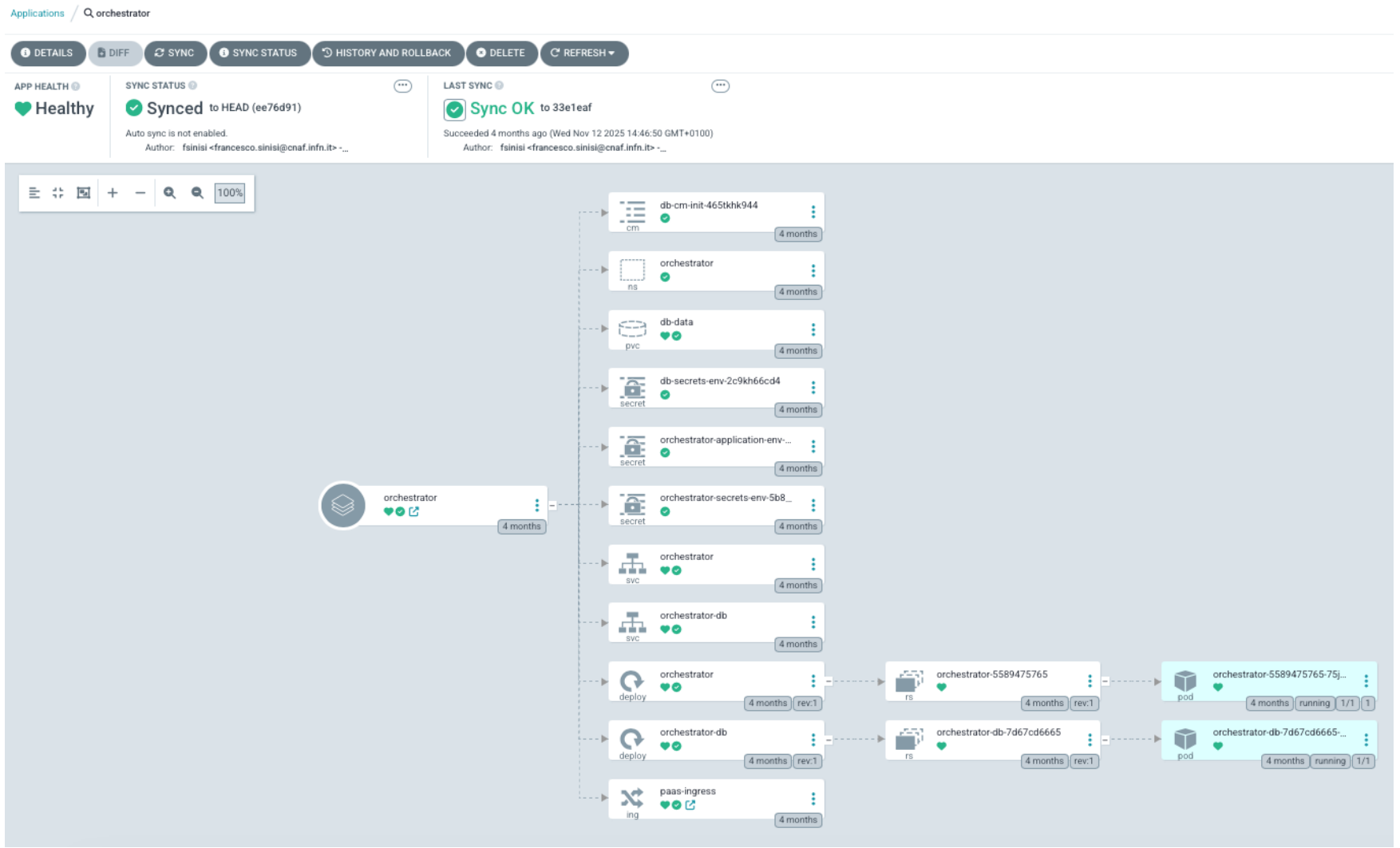
# Current status

- We deployed a **Kubernetes cluster in HA**, to improve resilience
- We integrated **INDIGO IAM** for users' authentication
- We traduced Ansible recipes, related to the services of the PaaS Orchestration system, in Kubernetes manifests
- We organized the manifests using **Kustomize**
  - Allows customization of configurations for different deployments
  - Uses bases + overlays to manage environment-specific changes
  - Reduces duplication and improves maintainability






It **facilitates service management** by allowing admins to check status, view logs, and more


Applications
orchestrator
DETAILS
DIFF
SYNC
SYNC STATUS
HISTORY AND ROLLBACK
DELETE
REFRESH
APP HEALTH
Healthy
SYNC STATUS
Synced to HEAD (ee76d91)
Auto sync is not enabled.
Author: fsinisi <francesco.sinisi@cnaf.infn.it> -...
LAST SYNC
Sync OK to 33e1eaf
Succeeded 4 months ago (Wed Nov 12 2025 14:46:50 GMT+0100)
Author: fsinisi <francesco.sinisi@cnaf.infn.it> -...
100%
orchestrator
db-cm-init-465tkhk944
orchestrator
db-data
db-secrets-env-2c9kh66cd4
orchestrator-application-env-...
orchestrator-secrets-env-5b8_...
orchestrator
orchestrator-db
orchestrator
orchestrator-db
paas-ingress
orchestrator-5589475765
orchestrator-db-7d67cd6665
orchestrator-5589475765-75j...
orchestrator-db-7d67cd6665-...
4 months
rev:1
running
1/1

# Conclusions and next steps

- Ongoing migration **from Ansible-based VM deployments to ArgoCD on Kubernetes**
- Adoption of a **GitOps** approach to improve automation, consistency, and traceability
- Refactoring existing manifests and creating **Kustomize** bases and overlays
- Enabling environment-specific configurations for different deployments
- This activity is part of a **broader refactoring of the PaaS Orchestration system**
- Next step: complete Kustomize structure and integrate applications with ArgoCD

# Thank you

luca.giommi@cnaf.infn.it