

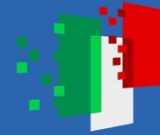
Deploying the INDIGO PaaS Orchestration System Components on a Kubernetes-Based Infrastructure Using ArgoCD

L. Giommi – INFN CNAF

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

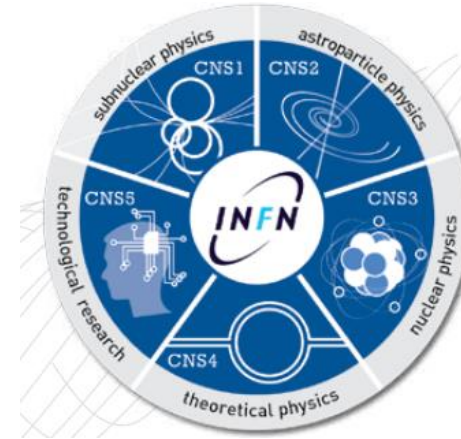


International Symposium on Grids & Clouds (ISGC) 2026 | 15-20 March 2026



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

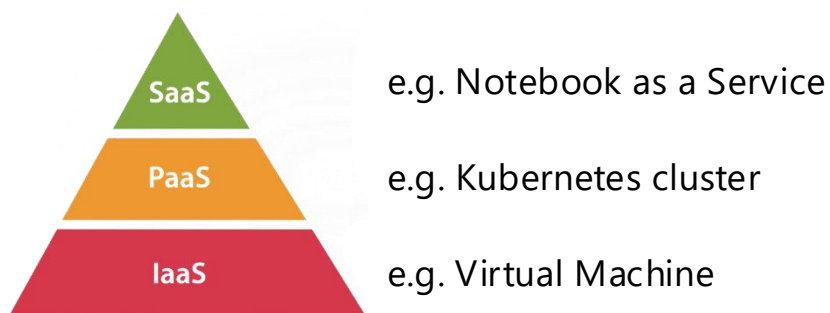


The INFN Cloud ecosystem

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**





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** (Topology 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

The screenshot shows the PaaS Orchestrator Dashboard. On the left is a dark blue sidebar with the INFN logo and the title 'Dashboard'. The sidebar contains a navigation menu with 'HOME', 'DEPLOYMENTS', 'ADVANCED', 'EXTERNAL LINKS', and 'ADMIN'. At the bottom of the sidebar, there are 'Info' and 'Help' links, and a user profile for 'Luca Giommi' with the email 'admins/catchall'. The main content area is white and features a 'REPORT' section with three status indicators: 'CREATION COMPLETED' (2), 'CREATION IN PROGRESS' (0), and 'CREATION FAILED' (2). Below this is a 'SERVICES' section with a search bar. Under 'CENTRALISED SERVICES', there are two service cards: 'INFN Cloud object storage' (described as 'the centrally managed service based on Ceph Rados-Gateway') and 'Notebooks as a Service (NaaS)' (described as 'Jupyter Notebooks as a Service'). Each card includes a logo, a description, and a 'GO TO SERVICE' button.

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



Deploy a PaaS service

ON-DEMAND SERVICES



Virtual machine

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



CONFIGURE →

Scheduling

SCHEDULING TYPE

- Automatic
- Manual

Select a deployment provider or let the system choose automatically

PROVIDER

- ✓ RECAS-BARI: org.openstack.nova
- BACKBONE - bari: org.openstack.nova
- BACKBONE - cnaf: org.openstack.nova
- CLOUD-CNAF-T1: org.openstack.nova

CANCEL ↻

Virtual machine

STEP 3/4

DEPLOYMENT DESCRIPTION (0/50)

Description

CONFIGURATION ADVANCED

HOSTNAME

vnode0

Hostname

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

CANCEL ↻

← Back

CONTINUE →



Features of the PaaS Orchestrator dashboard

My deployments

Refresh New deployment +

Show 10 entries Show deleted deployments Search:

DEPLOYMENT IDENTIFIER	DESCRIPTION	STATUS	CREATION AT	DEPLOYED AT	REGION	Actions
11eecd1-1dd0-ac5f-8be4-56fce75e0bfa	MLaaS Giommi	CREATE_COMPLETE	2024-02-16 13:41:00	CLOUD-CNAF-T1	tier1	<ul style="list-style-type: none"> Details Retry Show template Log Manage Nodes Delete

Showing 1 to 1 of 1 entries

User view

Virtual Nodes

← Back Refresh Add Node +

Show 10 entries Search:

NAME	HARDWARE CONFIGURATION	NETWORK INTERFACES	STATUS	ACTIONS
k8s-master-server-ffe0239a-1aaf-11f0-a32a-fa163e537d21	cores: 2 ram: 3906.25 MB disk: 18.62645149230957 GB Operating System: debian 12	net_interface_1_ip: 131.154.99.243 net_interface_0_ip: 192.168.12.91	STARTED	<ul style="list-style-type: none"> Stop Start Delete
k8s-node-server-fe5ad1aa-1aaf-11f0-a743-fa163e537d21	cores: 2 ram: 3906.25 MB disk: 37.25290298461914 GB Operating System: debian 12	net_interface_0_ip: 192.168.12.221	STARTED	<ul style="list-style-type: none"> Stop Start Delete

Deployments full list

Show 10 entries Show deleted deployments Search:

DEPLOYMENT IDENTIFIER	DESCRIPTION	STATUS	USER	CREATION TIME	DEPLOYED AT	REGION	GROUP	Actions
11f02cdd-1bbd-2395-8ecb-02424a612ab9	iam-dev	CREATE_COMPLETE	017d3540-a151-464e-bf13-fc7152bb7088	2025-05-09 13:54:00	BACKBONE	bari	admins/training	<ul style="list-style-type: none"> Details Show template Log Manage Ports Manage Nodes Delete
11f02cdb-bf5c-df70-8ecb-02424a612ab9	iam-dev	CREATE_FAILED	017d3540-a151-464e-bf13-fc7152bb7088	2025-05-09 13:44:00	BACKBONE	bari	admins/training	
11f02cb3-a61a-d648-8ecb-02424a612ab9	test VM retry_2	CREATE_COMPLETE	564f8033-4025-4fad-889f-83d01fec157c	2025-05-09 08:57:00	BACKBONE	bari	admins/beta-testers	

Admin view

Deployments Overview

Refresh Provider Group

Deployments status

Groups

Providers

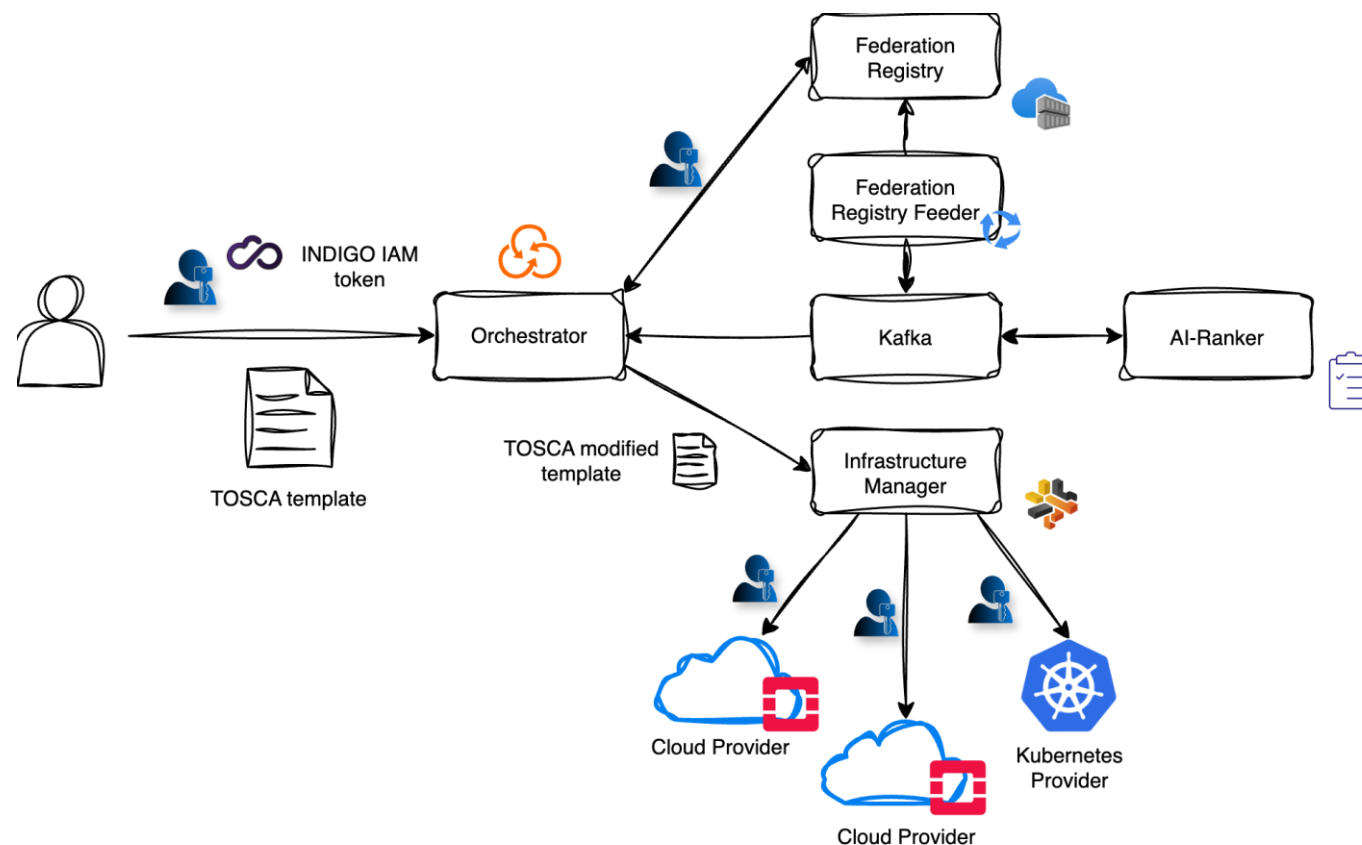
Templates Usage

Show 10 entries Search:

TEMPLATE NAME	INSTANCES
single-vm/single_vm.yaml	91
single-vm/single_vm_with_volume.yaml	75
kubernetes/k8s_cluster.yaml	46
jupyter/jupyter_vm.yaml	43

The current INDIGO PaaS Orchestration system of INFN Cloud

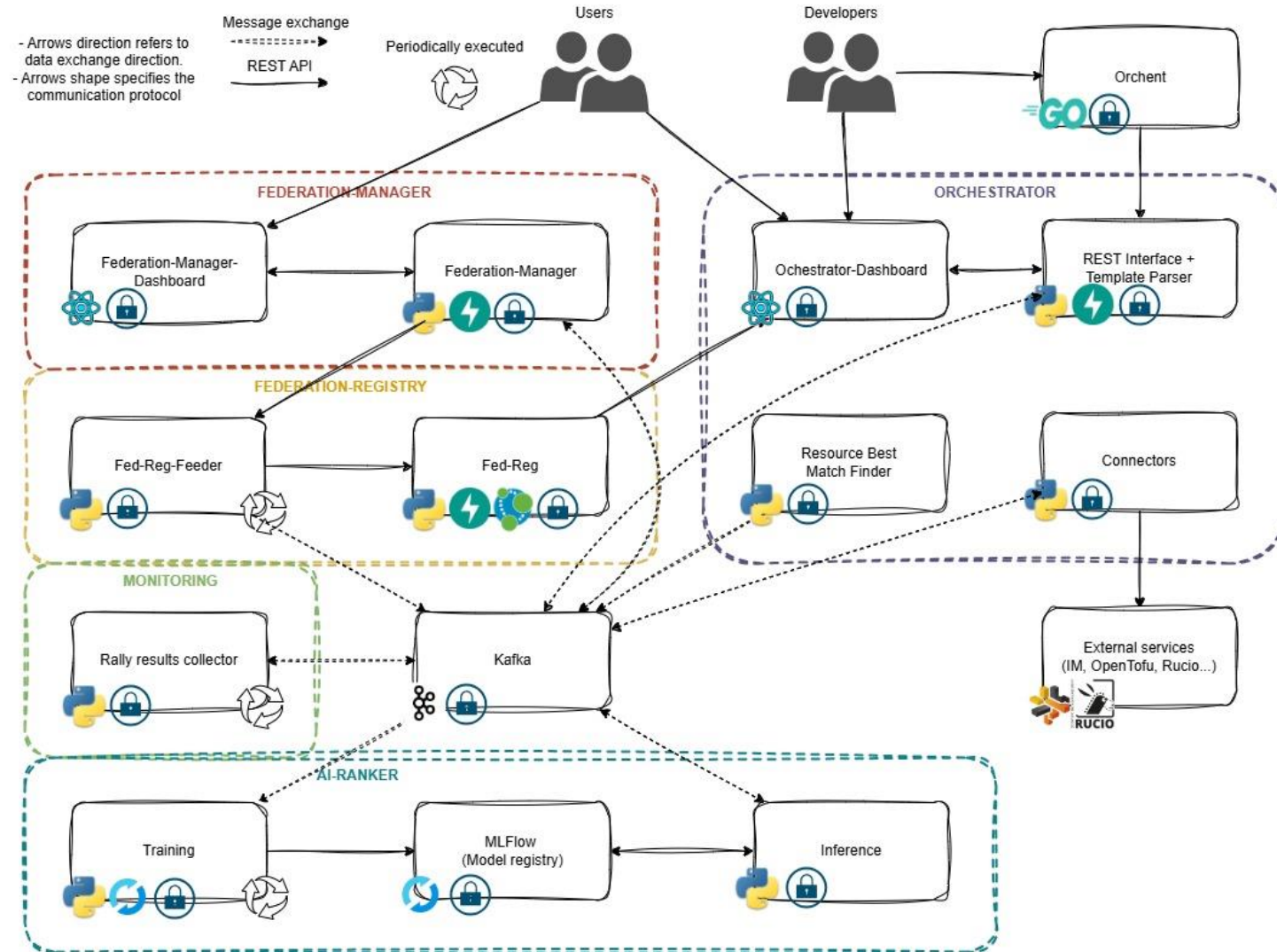
- The federative middleware of INFN Cloud is based on the **INDIGO PaaS Orchestration system**, consisting of interconnected open-source microservices
- The Orchestrator receives high-level deployment requests in the form of TOSCA templates and coordinates the deployment process by using the **Infrastructure 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





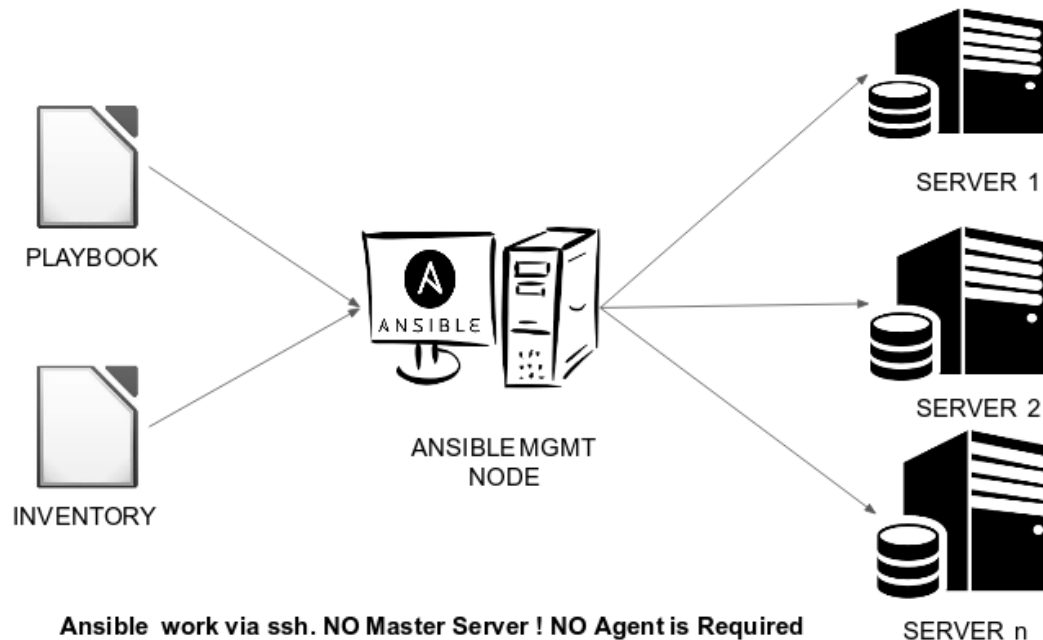
The new PaaS architecture

- 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



Welcome to **inf-n-cloud**

Sign in with



Not a member?

Apply for an account



The screenshot shows the Argo CD web interface. On the left is a dark sidebar with navigation options: Applications, Settings, User Info, and Documentation. Below this are filters for Favorites Only, SYNC STATUS (Unknown, Synced, OutOfSync), HEALTH STATUS (Progressing, Suspended, Healthy, Degraded, Missing, Unknown), and LABELS. The main content area is titled 'Applications' and features buttons for '+ NEW APP', 'SYNC APPS', and 'REFRESH APPS', along with a search bar. A modal window displays details for an application named 'orchestrator': Project: default; Labels: (empty); Status: Healthy Synced; Repository: https://baltig.infn.it/inf-n-cloud/wp5/paa...; Target R...: HEAD; Path: orchestrator-java/kustomize/overlays/p...; Destination: in-cluster; Namespace: orchestrator; Created: 11/11/2025 18:35:35 (4 months ago); Last Sync: 11/12/2025 14:46:50 (4 months ago). At the bottom of the modal are buttons for SYNC, REFRESH, and DELETE.

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



Applications / Q.orchestrator

DETAILS DIFF SYNC SYNC STATUS HISTORY AND ROLLBACK DELETE REFRESH

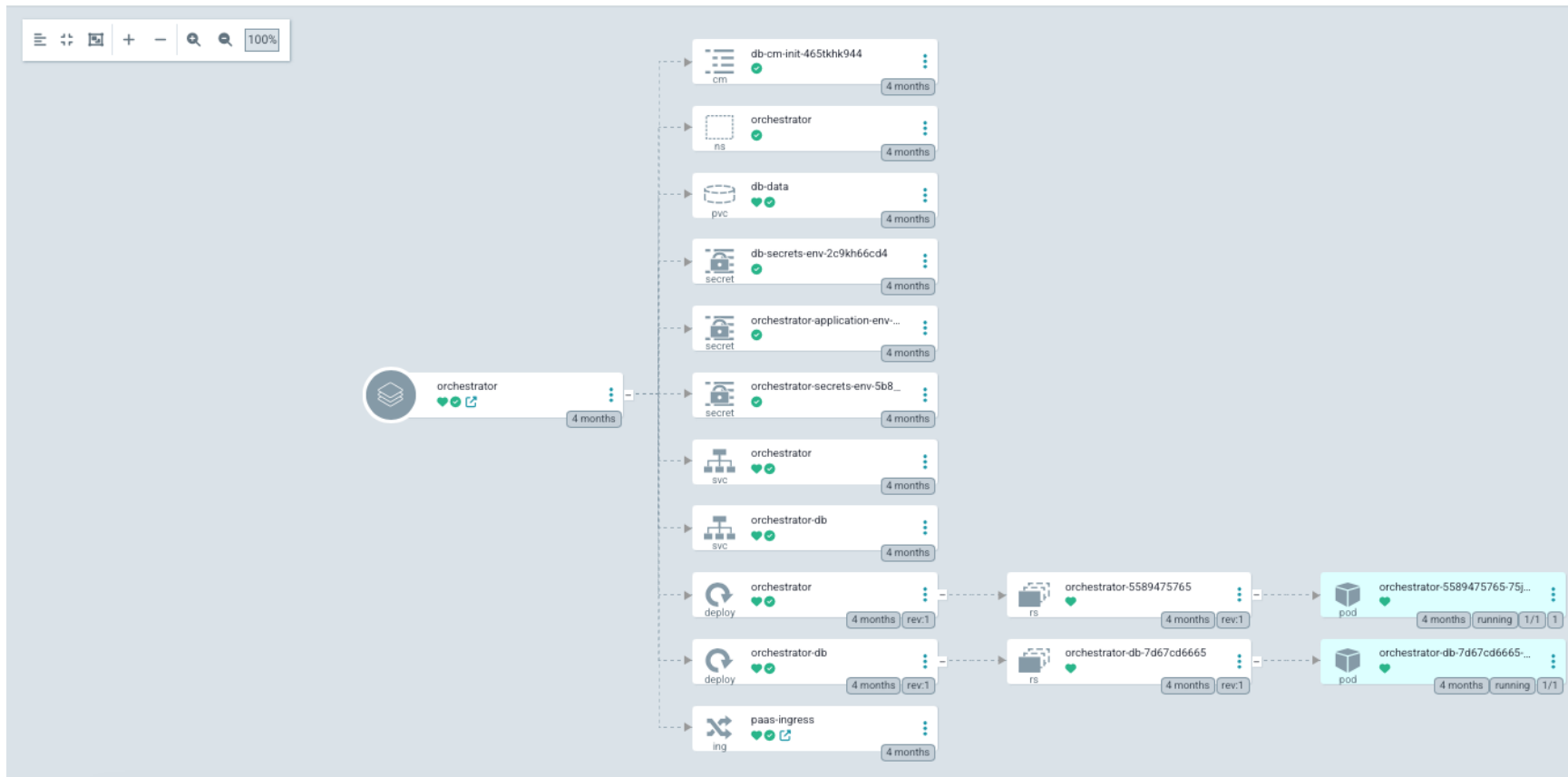
APP HEALTH
Healthy

SYNC STATUS
Synced to HEAD (ee76d91)

Auto sync is not enabled.
Author: fainisi <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: fainisi <francesco.sinisi@cnaf.infn.it> - ...



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



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Thank you

luca.giommi@cnaif.infn.it

