

# Introduzione al deploy di un'Analysis Facility (AF)

Gianluca Sabella, Elvira Rossi, Bernardino Spisso

**Meeting WP5 Spoke2**

22/03/2023

## Obiettivo

Rendere disponibili alla comunità scientifica e all'utenza in generale le risorse di calcolo del progetto, in modo

- rapido
- efficace
- tale da non impattare in modo troppo oneroso sugli amministratori delle risorse

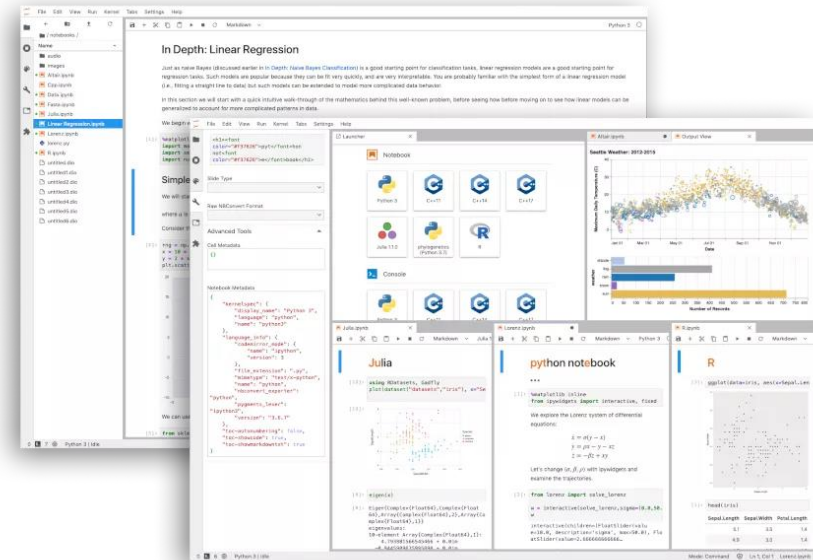


**E' NECESSARIO UN SISTEMA DINAMICO, SCALABILE E RAPIDO, PER IL DEPLOY DI UNA INFRASTRUTTURA DI CALCOLO RIVOLTA AI SINGOLI UTENTI**

# Analysis Facility (AF)

L' AF si prefigge di dare una risposta alle precedenti richieste tramite l'utilizzo di diverse tecnologie. Tra queste:

- Jupyter Hub/Lab: ambiente di sviluppo interattivo, basato sul Web, in grado di gestire accessi multipli



## Analysis Facility (AF)

L' AF si prefigge di dare una risposta alle precedenti richieste tramite l'utilizzo di diverse tecnologie. Tra queste:

- Jupyter Hub/Lab: ambiente di sviluppo interattivo, basato sul Web. In grado di gestire accessi multipli
- DASK: libreria Python open-source e flessibile per il calcolo parallelo



Semplifica la scalabilità delle librerie Python (NumPy, Panda, ecc. ...)



Permette di parallelizzare codice Python

## Analysis Facility (AF)

L' AF si prefigge di dare una risposta alle precedenti richieste tramite l'utilizzo di diverse tecnologie. Tra queste:

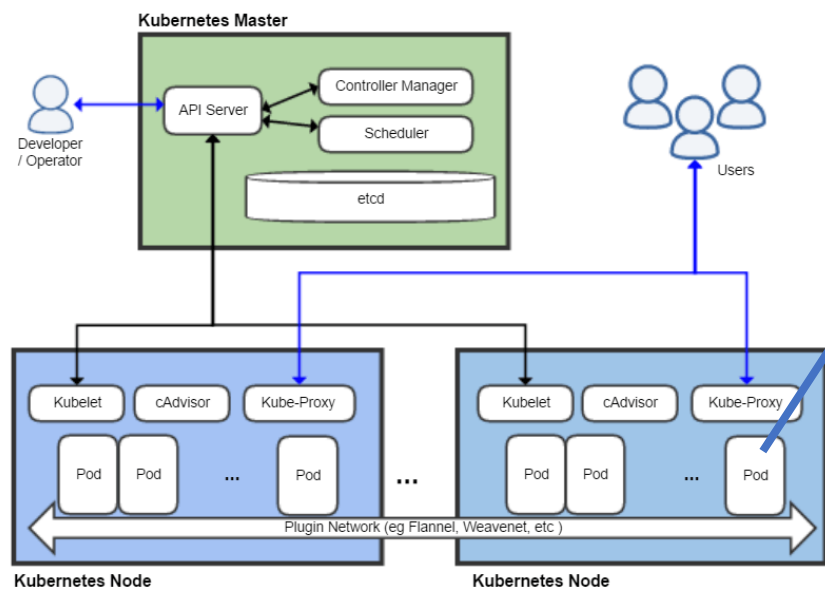
- Jupyter Hub/Lab: ambiente di sviluppo interattivo, basato sul Web. In grado di gestire accessi multipli
- DASK: libreria Python open-source e flessibile per il calcolo parallelo
- Kubernetes: sistema open source di gestione e orchestrazione di Containers



**kubernetes**

## Analysis Facility (AF)

Già sono in corso delle sperimentazioni su deploy che sfruttano la tecnologia dei containers orchestrati da Kubernetes (D. Ciangottini & D. Spiga) che permettono l'installazione dell'AF in modo rapido e scalabile



POD: unità di calcolo distribuibile più piccola → gruppo di uno o più container

Ogni container è standard e ripetibile → si ottiene lo stesso comportamento ovunque lo si esegue



Facilità e velocità di aggiunta di nodi



**SCALABILITA'**

## Attività in corso e in programma a Napoli

Step dell'attività:

1. Riproduzione del deploy localmente → IN CORSO
2. Test sul deploy locale
3. Adattamento del deploy ad infrastrutture cloud (INFN Cloud)

## Stato attuale

Fino ad ora, nessuna infrastruttura presso la sede di Napoli utilizzava lo strumento Kubernetes

Attualmente, stiamo preparando una infrastruttura servendoci degli strumenti descritti per installare Kubernetes

Risorse a disposizione per un test immediato:

- 2 macchine virtuali identiche istanziate tramite Proxmox (per ora con risorse minimali che all'occorrenza possono essere potenziate a seconda della necessità)
- SO Rocky Linux 8.6



## Passi successivi e finali

Obiettivi a breve termine

Testare il deploy dell'AF su  
infrastruttura locale

Obiettivi a medio termine

Testare il deploy dell'AF su  
infrastrutture cloud locali  
(deploy di Kubernetes su Open  
Stack)

Testare l'utilizzo di computing  
node muniti di GPU

Obiettivi a lungo termine

Traslare il deploy,  
opportunamente testato,  
sul cloud INFN a livello  
nazionale

## Tempistiche/Conclusioni

- Step0 (corrente): preparazione di una infrastruttura Kubernetes di test per il deploy [1 mese]
- Step1: testing effettivo del deploy sulla predetta infrastruttura [2-3 mesi]
- Step2: adattamento dell'intera infrastruttura (Kubernetes + AF su open stack) [3-4 mesi]
- Step3: passaggio su cloud nazionale [?]