



DIE
TI. UNI
NA



Istituto Nazionale di Fisica Nucleare



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Le attività del T2 di Napoli con Kubernetes: il caso d'uso delle analysis facilities

Gianluca Sabella, Bernardino Spisso

Problema: individuare un *tradeoff* tra rapidità e flessibilità del deploy ed efficienza di utilizzo delle risorse e performance dei servizi

Esempio tipico è fornire infrastruttura di calcolo agli utenti

Soluzione in corso di sperimentazione a Napoli: l'utilizzo della tecnologia dei container che eseguono le applicazioni e dello strumento Kubernetes per l'orchestrazione

Scaletta

- Breve overview di Kubernetes
- Strumenti da noi utilizzati per l'installazione e la gestione del cluster Kubernetes
- Descrizione del caso d'uso dell'Analysis Facility



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA

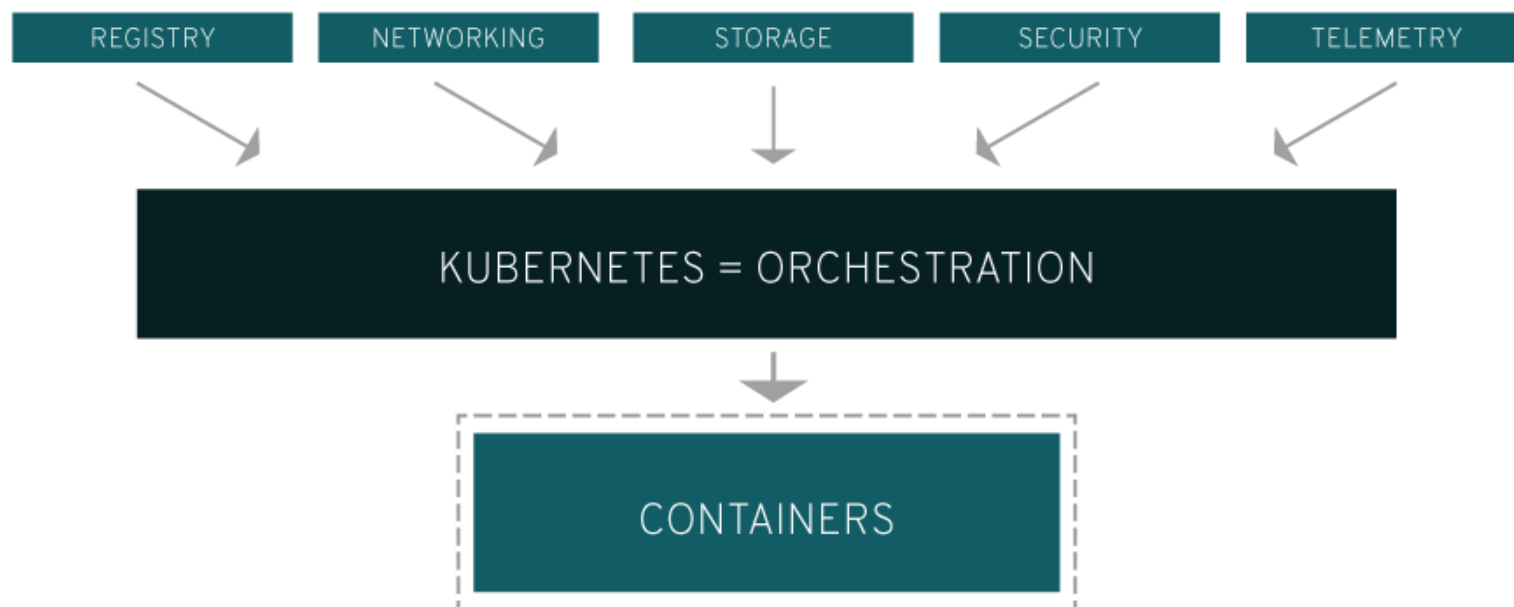


Kubernetes: Orchestrazione del Cluster per la Distribuzione Scalabile

Kubernetes è un sistema open-source per l'orchestrazione e la gestione automatizzata di container applicativi.

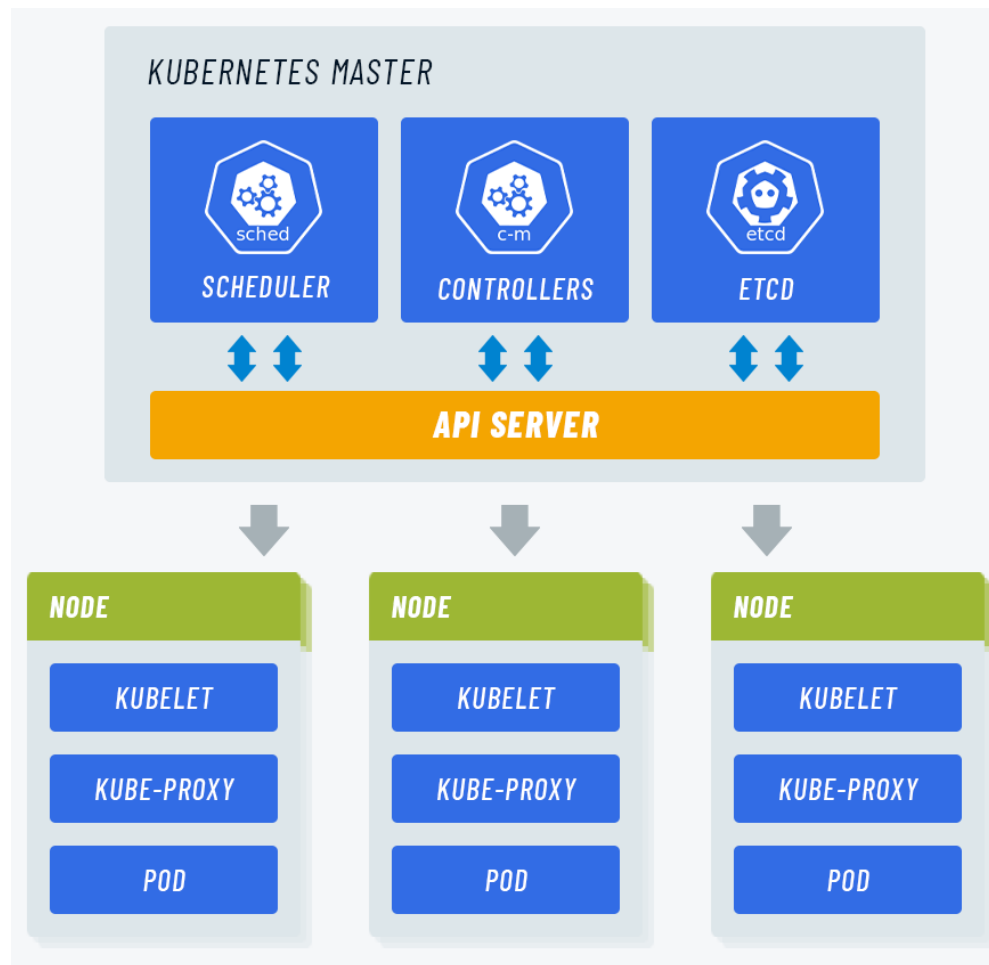
Scopo di Kubernetes:

- semplifica le operazioni di deployment, aggiornamento e gestione delle applicazioni, fornendo un'infrastruttura resiliente e flessibile.
- facilita l'organizzazione e la gestione delle risorse di elaborazione, archiviazione e rete all'interno di un cluster.



L'Architettura di Kubernetes

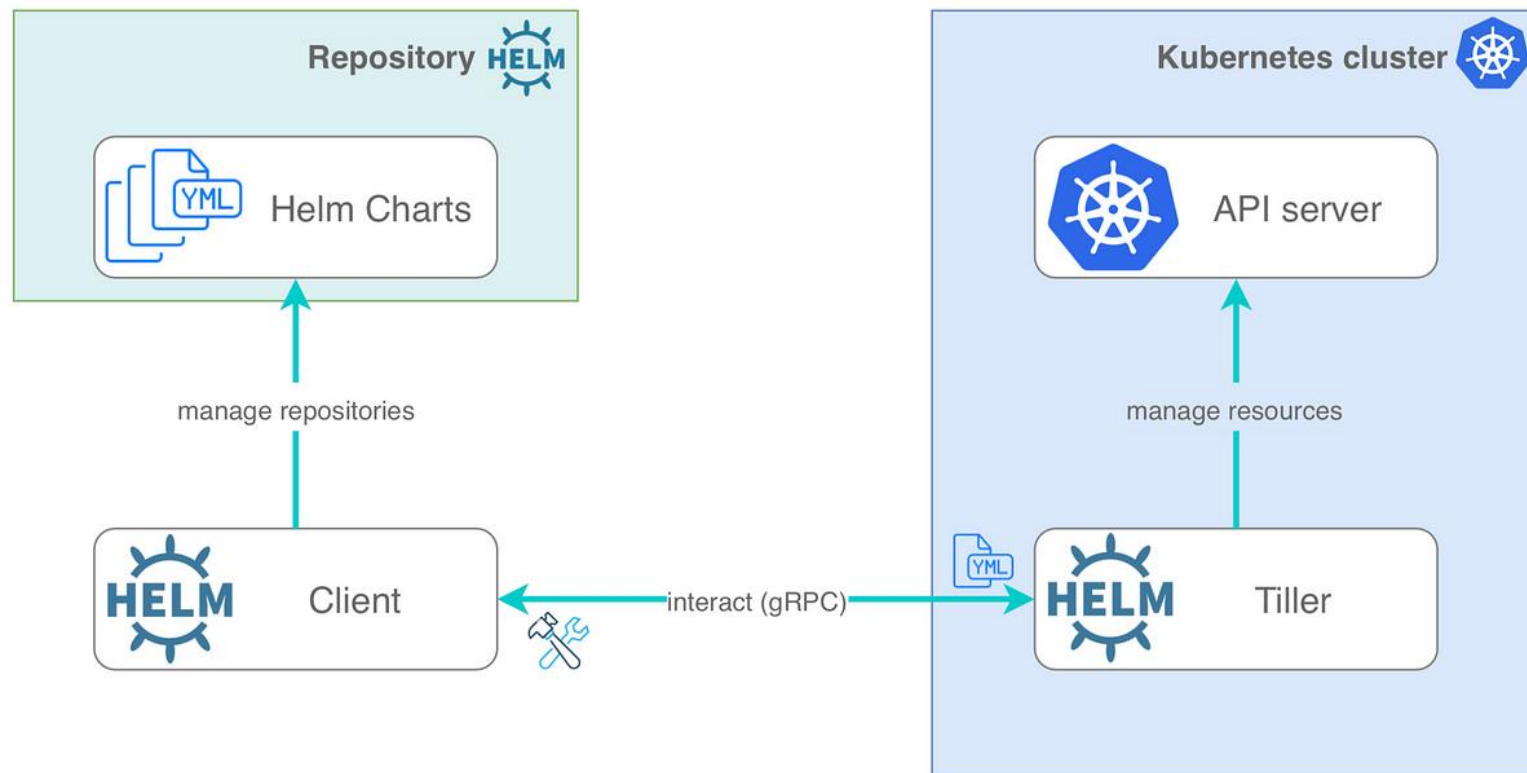
- **Master Node**
ha il compito di coordinare e gestire l'intero cluster Kubernetes. È costituito da vari componenti, tra cui l'API Server, il Controller Manager, lo Scheduler e l'etcd.



- **Worker Nodes**
rappresentano i nodi del cluster in cui i container vengono eseguiti. Ogni Worker Node ospita il Kubelet, il Kube-proxy e i container stessi, fornendo la capacità computazionale effettiva del cluster.

Strumenti utilizzati (1): HELM Client

Helm è un **packet manger** che facilita la gestione delle applicazioni Kubernetes attraverso la definizione, l'installazione e la gestione dei pacchetti chiamati **chart**.



I chart di Helm sono pacchetti preconfigurati che contengono tutte le risorse necessarie per eseguire un'applicazione su Kubernetes, inclusi servizi, deployment, ingress, configurazioni, variabili, ecc. ...



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Strumenti utilizzati (2): Rancher Kubernetes Engine (RKE) **RANCHER** BY SUSE

RKE è una distribuzione Kubernetes certificata che viene eseguita interamente all'interno dei container Docker.

Risolve la complessità dell'installazione con Kubernetes rimuovendo la maggior parte delle dipendenze host e presentando un percorso stabile per la distribuzione, gli aggiornamenti e i rollback.

Requisito: utilizzo di un tool per
l'interazione con il cluster
Kubernetes



**Il command line tool
Kubectl**

Generazione cluster Kubernetes: RKE + YAML

YAML è un linguaggio di programmazione leggibile in chiaro per la serializzazione di dati che viene spesso impiegato per la scrittura dei file di configurazione.

Il settaggio della configurazioni del cluster Kubernetes è avvenuta tramite file YAML (.yml o .yaml), utilizzato da RKE per la generazione del cluster (\$ rke up)

Anche i chart di HELM sono in formato yaml

```
helm install -n nginx --create-namespace nginx ingress-nginx/ingress-nginx --values ./manifests/helm/nginx-values.yaml
```

NGINX: web server open source utilizzato anche come proxy inverso, cache HTTP e bilanciatore di carico.

```
cluster_name: mycluster
kubernetes_version: v1.23.7-rancher1-1

resolv-conf: /etc/resolv.conf
kube-apiserver-arg:
  - kubelet-preferred-address-types=InternalIP

network:
  plugin: calico

services:
  kube-controller:
    extra_args:
      cluster-signing-cert-file: /etc/kubernetes/ssl/kube-ca.pem
      cluster-signing-key-file: /etc/kubernetes/ssl/kube-ca-key.pem
  kubelet:
    extra_binds:
      - /var/openeks/local:/var/openeks/local

addon_job_timeout: 300

ingress:
  provider: none

nodes:
  - address:
    user: almalinux
    role:
      - controlplane
      - etcd
      - worker
  - address:
    user: rke
    role:
      - worker
  - address:
    user: rke
    role:
      - worker
  - address:
    user: rke
    role:
      - worker
  - address:
    user: rke
    role:
      - worker
  - address:
    user: rke
    role:
      - worker
```



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Workflow

Passo 1 → Preparazione delle nuove macchine

Passo 2 → Installazione di Kubernetes sul nodo master, sfruttando RKE

Passo 2.5 → Installazione del tool «kubectl» di Kubernetes

Passo 3 → Installazione del client HELM sul nodo master

Passo 4 → Installazione della piattaforma Docker su tutti i nodi



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Dettagli del test bed del cluster Kubernetes a Napoli

Il nostro cluster Kubernetes è caratterizzato da 6 machine virtuali

Il nodo 1 (master) svolge
i ruoli di control plane,
etcd, e worker



I nodi da 2 a 6
svolgono il solo ruolo
di worker

Il master è equipaggiato con:

- VCPU: 8 (1 fisica da 8 core)
- Disco: 50 Gigabytes
- RAM: 16 Gigabytes
- SO: Alma Linux 9.2

Ogni macchina worker è equipaggiata con:

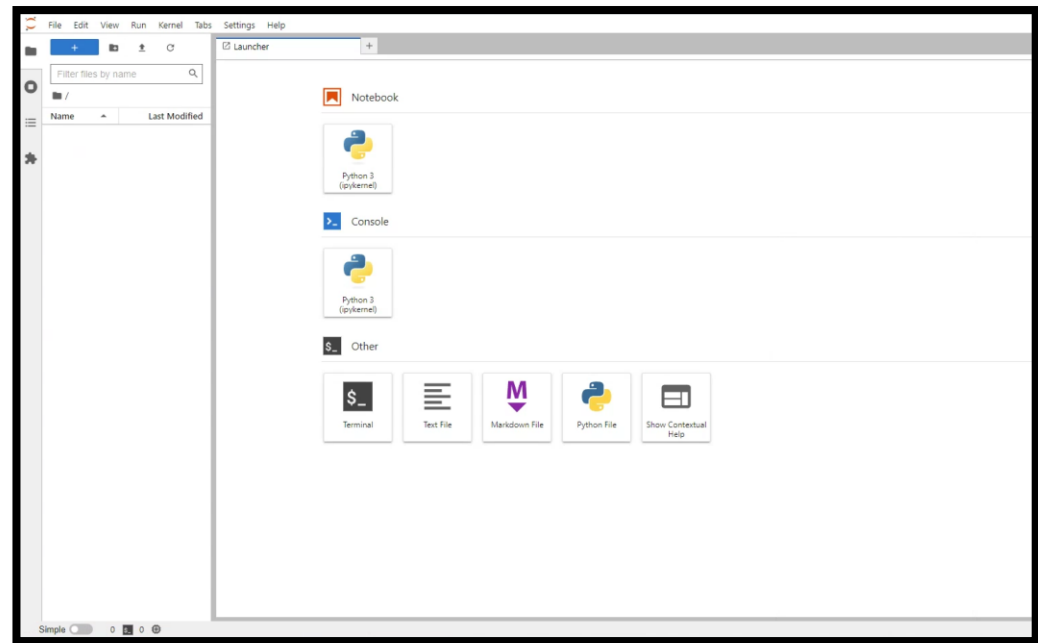
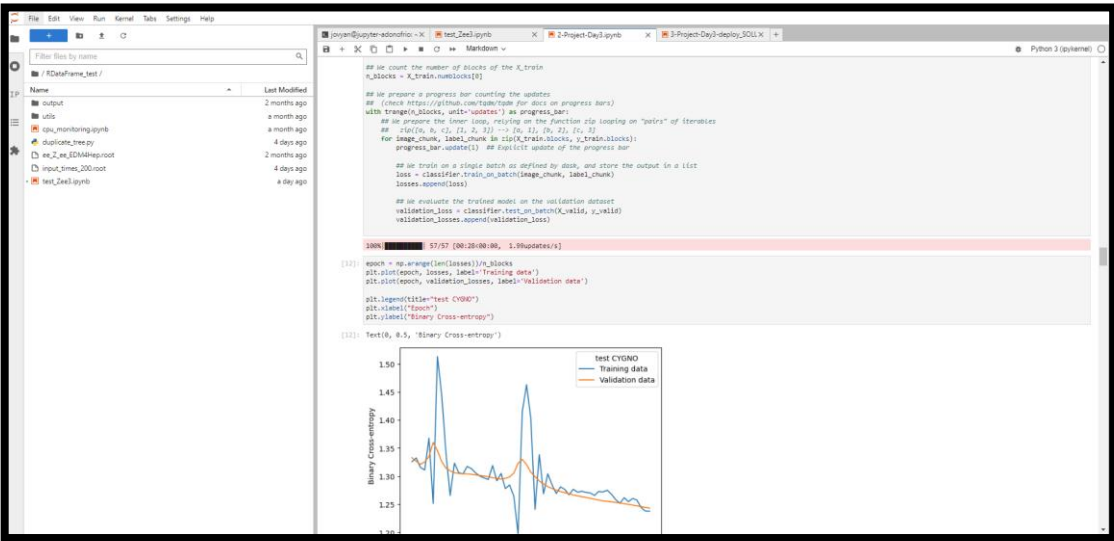
- VCPU: 32 (1 fisica da 32 core)
- Disco: 100 Gigabytes
- RAM: 64 Gigabytes
- SO: Alma Linux 9.2

Analysis Facility (AF)

```
kubectl apply -k
```

Tramite la predetta infrastruttura realizzata, attraverso Kubernetes abbiamo realizzato un prototipo per dare una risposta alle precedenti richieste il quale offre all'utente risorse di calcolo, utilizzando:

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



Analysis Facility (AF)

Tramite la predetta infrastruttura realizzata, attraverso Kubernetes abbiamo realizzato un prototipo per dare una risposta alle precedenti richieste il quale offre all'utente risorse di calcolo, utilizzando:

- **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 che può utilizzare i batch system più diffusi (SLURM, HTC Condor, ecc. ...) oppure è all'occorrenza fornito di uno scheduler nativo (la nostra scelta)

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

↓
Permette di parallelizzare codice
Python





Ministero
dell'Università
e della Ricerca



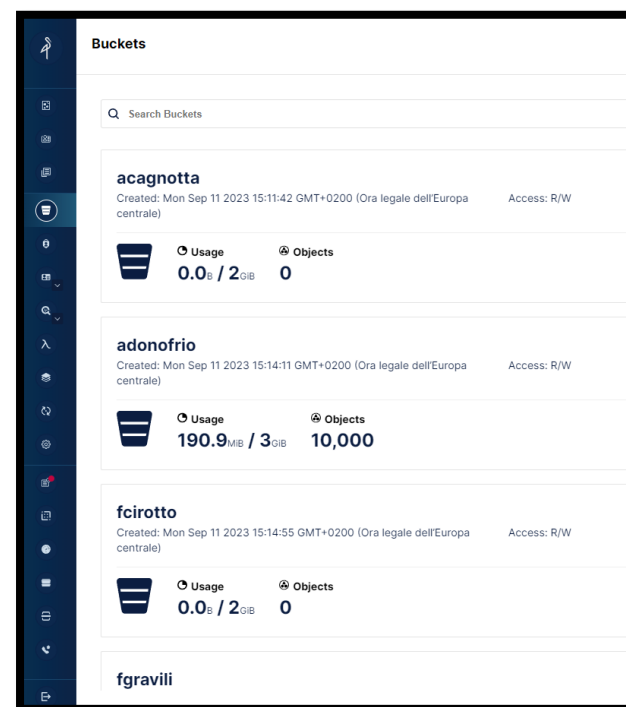
Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



Analysis Facility (AF)

Tramite la predetta infrastruttura realizzata, attraverso Kubernetes abbiamo realizzato un prototipo per dare una risposta alle precedenti richieste il quale offre all'utente risorse di calcolo, utilizzando:

- **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 che può utilizzare i batch system più diffusi (SLURM, HTC Condor, ecc. ...) oppure è all'occorrenza fornito di uno scheduler nativo (la nostra scelta)
- **MinIO**: Object Storage progettato per essere distribuito ovunque: cloud pubblico o privato, infrastruttura bare metal, ambienti orchestrati e infrastrutture edge.



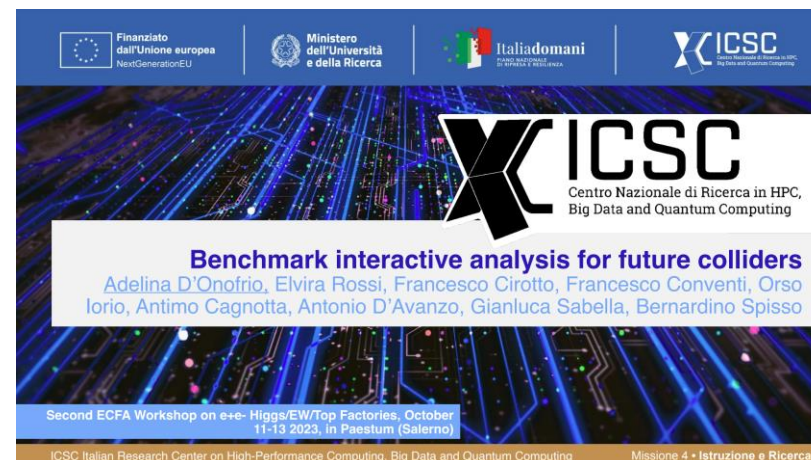
Conclusioni e passi futuri

In conclusione, stiamo testato la tecnologia di Kubernetes per fornire servizi di calcolo all'utenza, in modo rapido, scalabile e garantendo la portabilità delle applicazioni su vari ambienti e la loro gestione su differenti infrastrutture.

Conclusioni e passi futuri

In conclusione, stiamo testato la tecnologia di Kubernetes per fornire servizi di calcolo all'utenza, in modo rapido, scalabile e garantendo la portabilità delle applicazioni su vari ambienti e la loro gestione su differenti infrastrutture.

È stata già testata un'analisi reale come primo prototipo del progetto

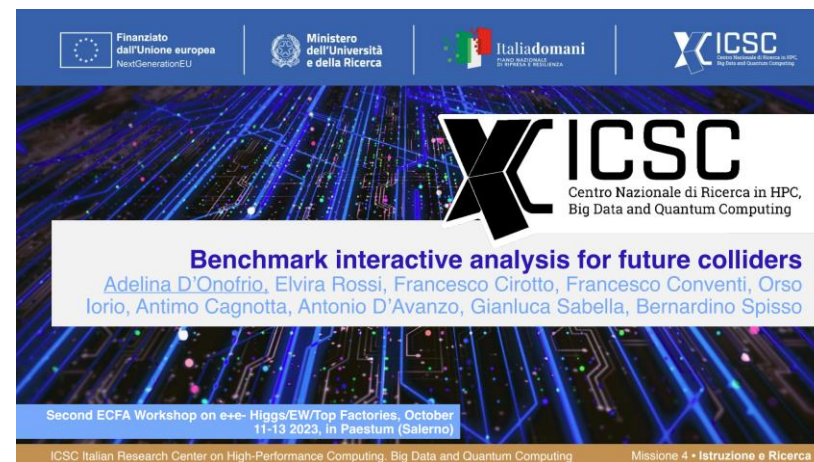


[Link qui](#)

Conclusioni e passi futuri

In conclusione, stiamo testato la tecnologia di Kubernetes per fornire servizi di calcolo all'utenza, in modo rapido, scalabile e garantendo la portabilità delle applicazioni su vari ambienti e la loro gestione su differenti infrastrutture.

È stata già testata un'analisi reale come primo prototipo del progetto



[Link qui](#)

PROSSIMI PASSI:

- La ricetta Kubernetes è in continuo miglioramento:
 - Stiamo testando la possibilità di fornire all'utente la scelta di diversi ambienti di calcolo
 - E' attualmente in corso la migrazione verso un sistema di riproduzione sull'INFN Cloud nazionale



Grazie per l'attenzione

Gianluca Sabella: gianluca.sabella@na.infn.it