



INFN Cloud: Use-case avanzati

Corso docker e orchestrazione di container.
Daniele Spiga - INFN-PG

Docker based (advanced) use cases



INFN Cloud Dashboard Deployments Advanced External Links Users ml-infn Daniele Spiga

Search...

Virtual machine	Virtual machine	Docker-compose
Run docker	Elasticsearch and Kibana	Apache Mesos cluster
Kubernetes cluster	Spark + Jupyter cluster	RStudio
TensorFlow with Jupyter	(Experimental) Jupyter with persistence for Notebooks	Computational environment for Machine Learning INFN (ML_INFN)
Computational environment for Machine Learning INFN (ONLY FOR TEST)	Galaxy	Working Station for CYGNO experiment

You can visit [here](#)

(if you are/once you get authorized)



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)
- 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
description

General IAM integration Advanced

jupyter_images
dodasts/mlinfm-base:v1 dodasts/mlinfm-conda-base:v2
Default image for jupyter server

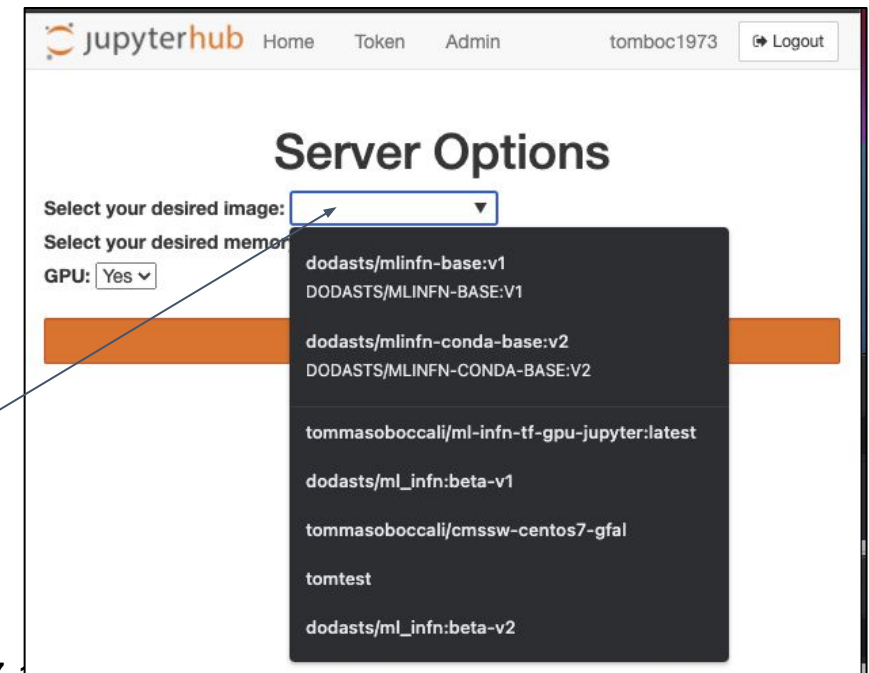
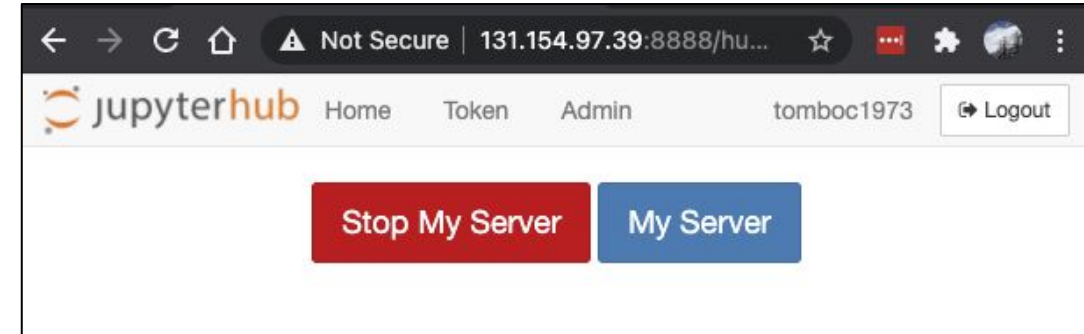
cvmfs_repos
cms.cern.ch sft.cern.ch atlas.cern.ch
CVMFS repositories to mount

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

Cosa gira nella VM?

- Un **jupyterhub** gira nella VM, e permette a **utenti autorizzati** di creare la loro istanza running mediante un container (preso o **localmente**, o direttamente da **dockerhub**)
- Tutti questi container usano le risorse della VM, che e' quindi shared per il gruppo di lavoro
- I container sono accessibili sia mediante Jupyter Notebooks, che via terminale (per il moment via browser, asap via ssh)
- L'amministratore può accedere alla VM sia ssh sia via browser



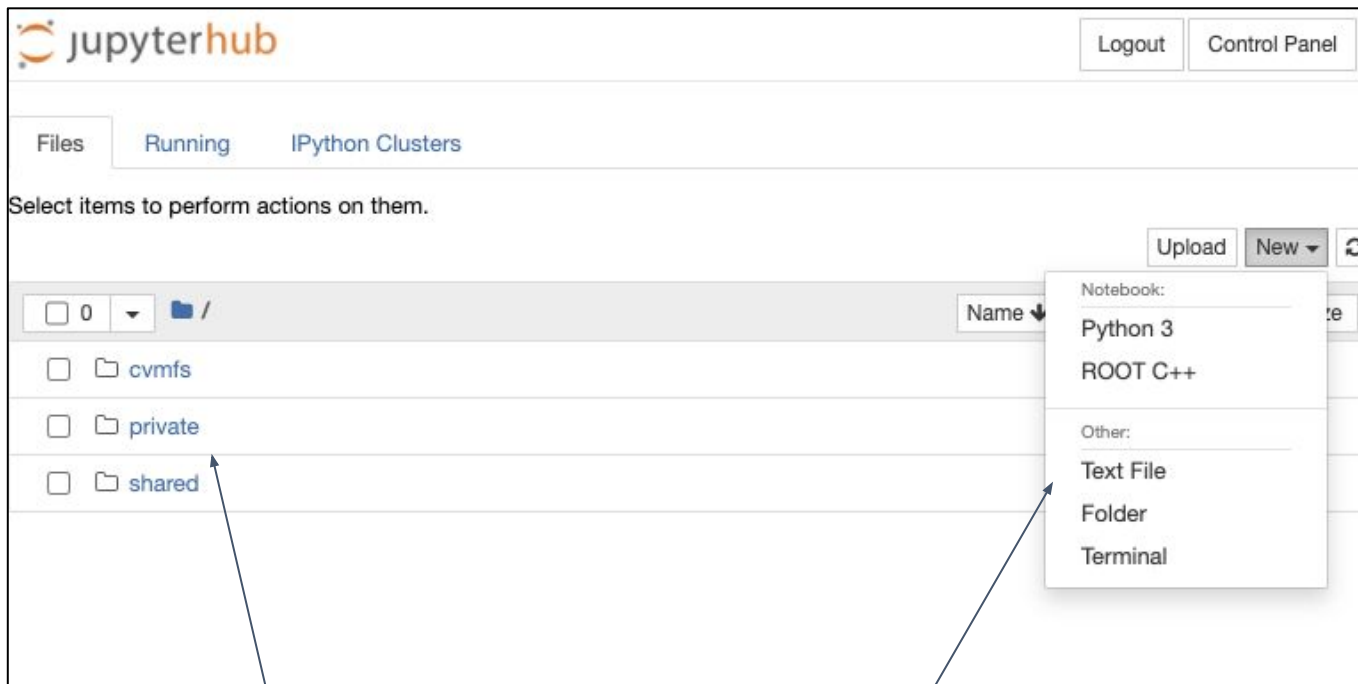
Qualunque container da dockerhub qui

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=0.0.0.0"           11 days ago   Up 11 days   0.0.0.0:8889->8889
e0f479af4a86   jupyterhub_backup_service          "cron -f"                             11 days ago   Up 11 days
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#
```


Accesso "utente"



Aree cvmfs e shared con tutti gli altri utenti della VM

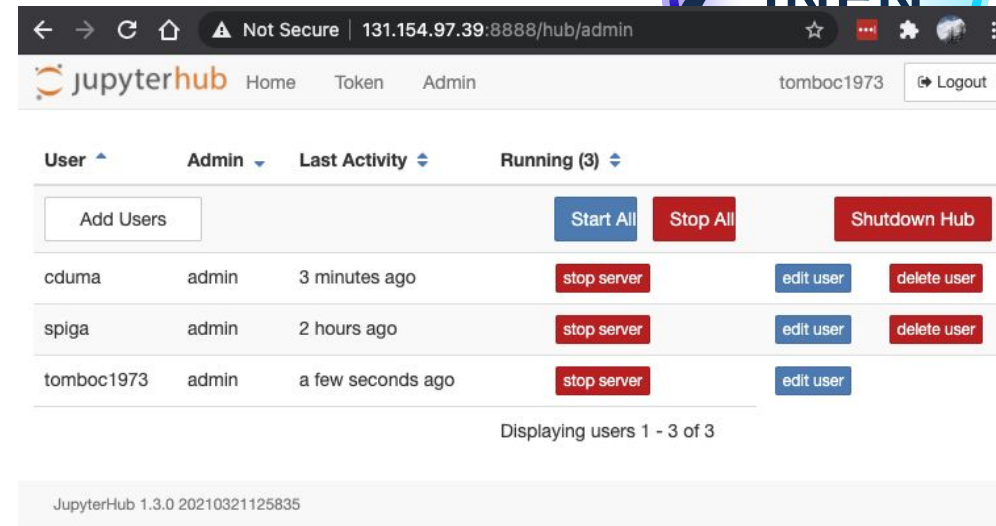
Accesso sia via notebooks che via terminal



Accesso terminale come root, 2 GPU visibili

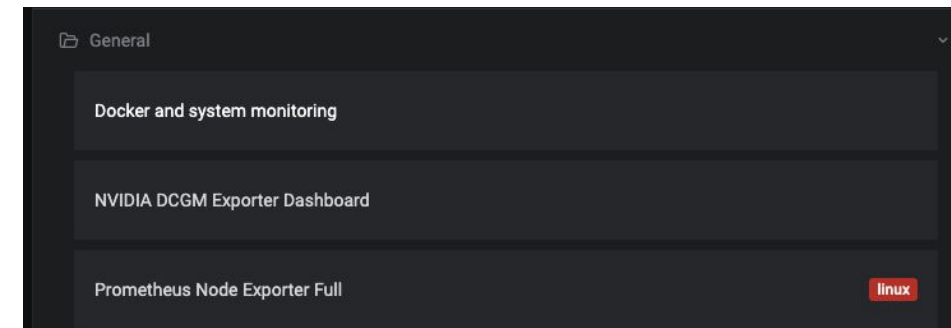
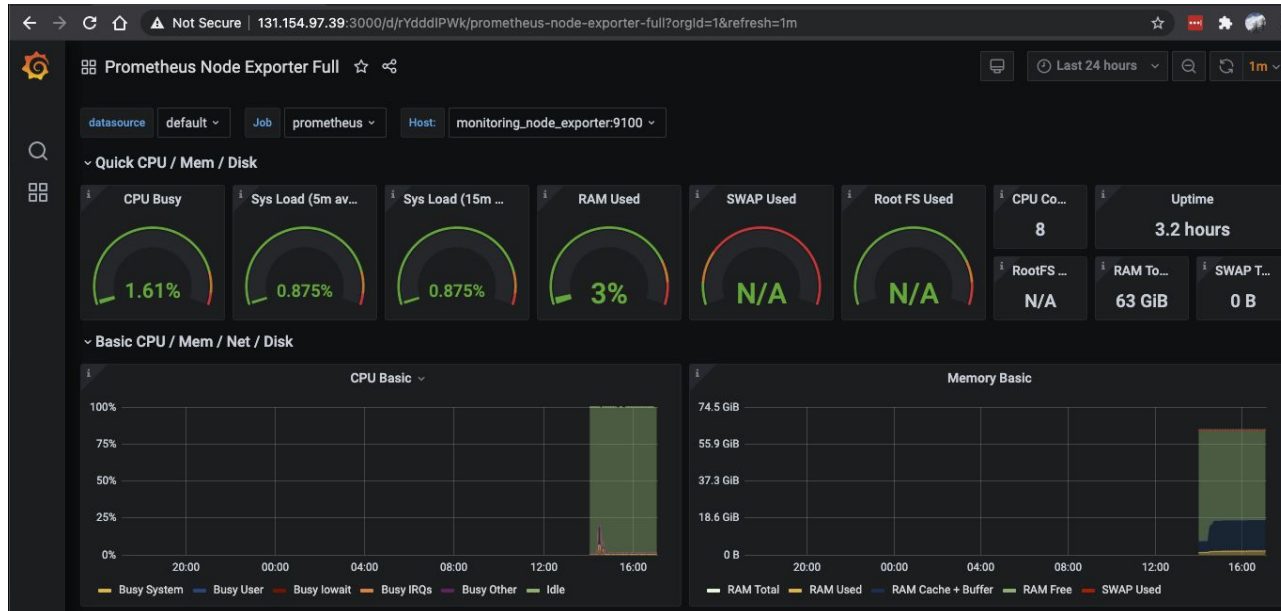
Monitoring etc

- L'amministratore può gestire i container
- Tutti possono vedere un monitoring dettagliato



A screenshot of the JupyterHub admin interface. The browser address bar shows '131.154.97.39:8888/hub/admin'. The page title is 'jupyterhub' with navigation links for 'Home', 'Token', and 'Admin'. The user 'tomboc1973' is logged in. The main content area shows a table of users with columns for 'User', 'Admin', 'Last Activity', and 'Running (3)'. There are buttons for 'Add Users', 'Start All', 'Stop All', and 'Shutdown Hub'. The table lists three users: 'cduma', 'spiga', and 'tomboc1973'. Below the table, it says 'Displaying users 1 - 3 of 3'. At the bottom, the version 'JupyterHub 1.3.0 20210321125835' is displayed.

User	Admin	Last Activity	Running (3)
cduma	admin	3 minutes ago	stop server, edit user, delete user
spiga	admin	2 hours ago	stop server, edit user, delete user
tomboc1973	admin	a few seconds ago	stop server, edit user



A screenshot of a configuration panel for 'General' settings. It lists several monitoring options: 'Docker and system monitoring', 'NVIDIA DCGM Exporter Dashboard', and 'Prometheus Node Exporter Full'. The 'Prometheus Node Exporter Full' option is highlighted with a red 'linux' label.

Something more advanced



As you saw there are several use cases even more “advanced” from the infrastructural point of view.

Those are build mostly using the tools discussed during this training

- Docker, docker-compose, Kubernetes, Mesos etc

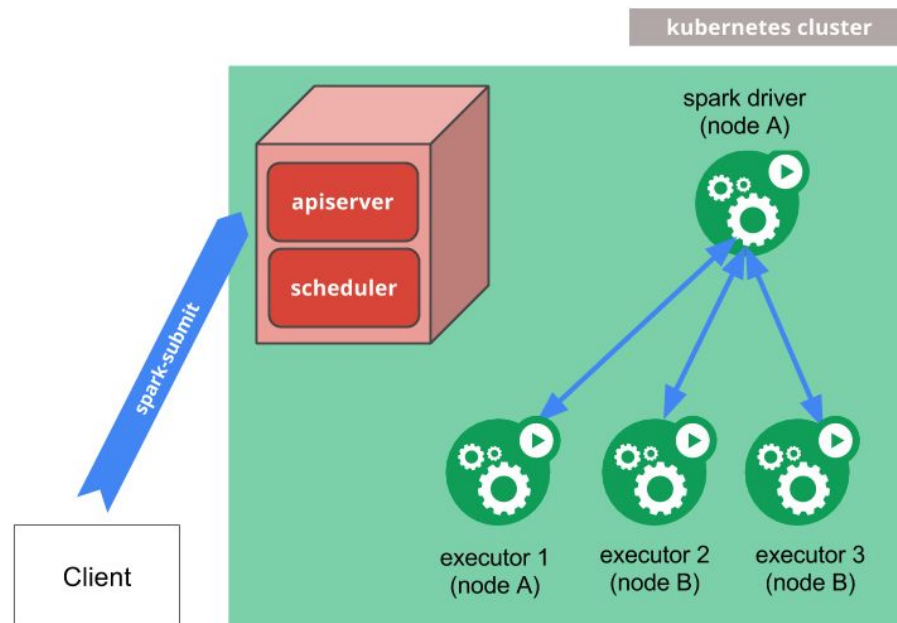
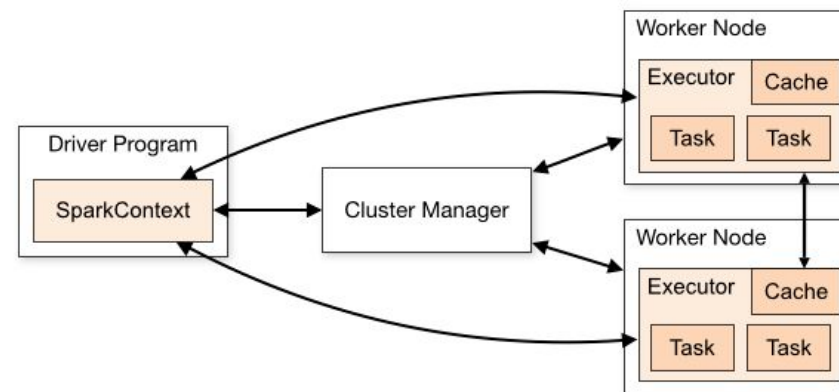
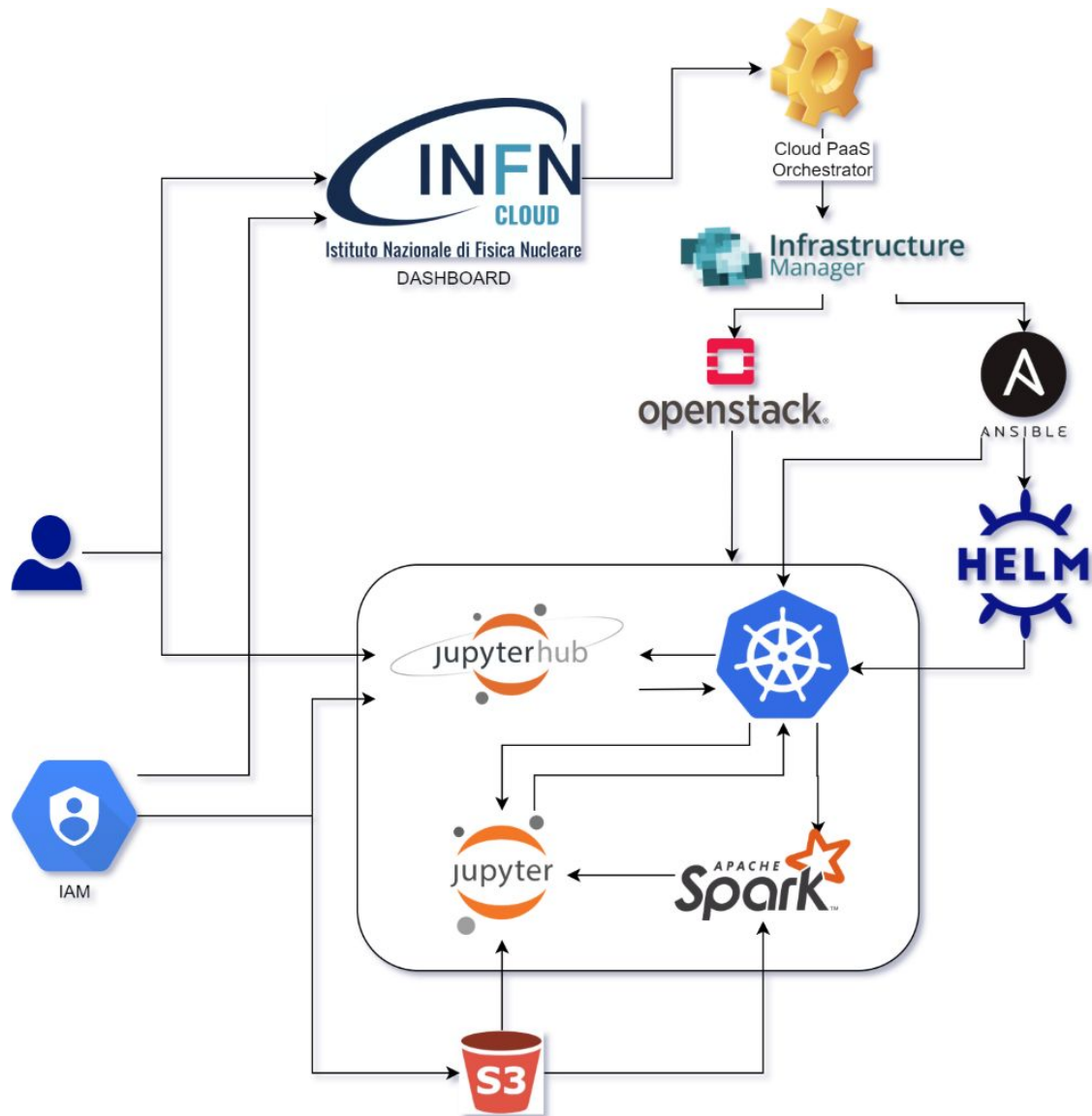
Two implementations that I would highlight here are

- Jupyter + Spark on top of Kubernetes
- HTCondor on top of Kubernetes



Jupyter + Spark + K8s

Schema

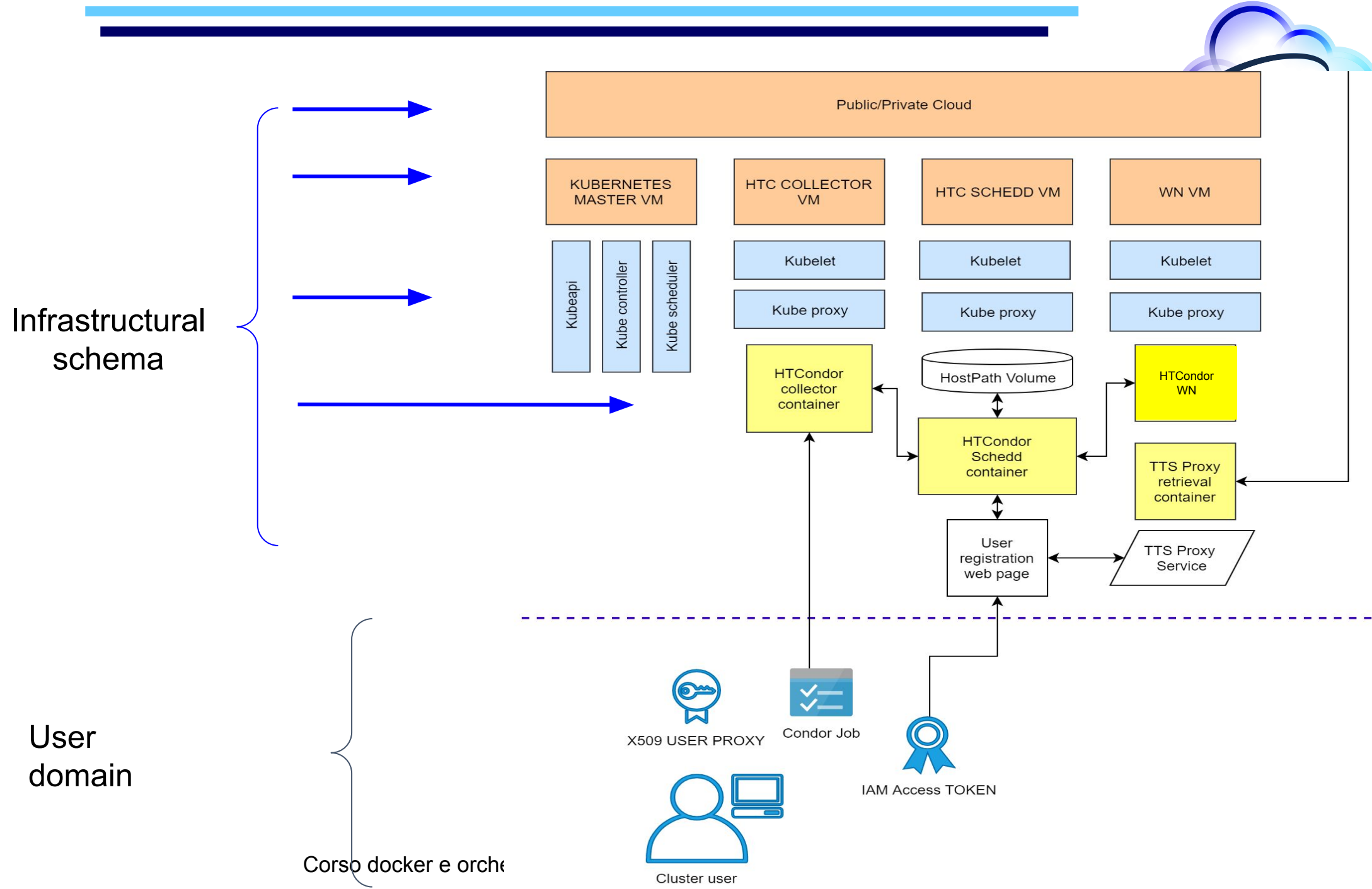


e di containers, 07-11/02 2022



Batch On Demand

Architectural Schema



HTCondor Dockerfile



Started few years ago, official containers not yet available we developed a different strategy

- Single image configured at runtime via ENV variable (to identify the role of the service)
 - Define ENV variables to configure daemons at runtime
- Plus several customisation such as a minimal Flask application to allow user registration (see later)
 - Need a condormapfile to allow user remote submission

<https://github.com/DODAS-TS/dodas-docker-images/blob/v0.1.0-condor/docker/htcondor/htcondor/Dockerfile>

- We migrated to the official images
 - Lighter, maintained ...
 - Adding our customizations

HTCondor on top of K8s: 6 key elements



- **Hosts:** K8s cluster requires at least 3+1 hosts
- **Topology:** Central Manager and Submit node are deployed on run on 2 dedicated hosts
- **Pods :** (“microservice”) 2 containers, HTCondor daemon come together with proxy manager (aka give me a X509 out of a incoming JWT)
- **Self healing:** e.g. run probes to check service status
- **Host selections:** Use k8s node labels and affinities are used to automate host selections
- **Spool directory:** a persistent volume and is mounted via PersistentVolume
 - This is somehow a first approach to the HA
- **HTCondor config Management:** HTCondor configuration dynamically managed through k8s configMap and secrets

Pods



This is a schedd. Two dockers run

```
containers:
  - name: tts
    image: 'dodasts/tts-cache:v0.1.3-k8s-11'
    args:
      - '--get-proxy'
      - '--period'
      - '120'
      - '--config'
      - /app/.config.yaml
    resources:
      requests:
        cpu: 100m
        memory: 500M
    volumeMounts:
      - name: proxydir
        mountPath: /root/proxy
      - name: uwdir
        mountPath: /home/uwdir
    terminationMessagePath: /dev/termination-log
    terminationMessagePolicy: File
    imagePullPolicy: IfNotPresent
  - name: schedd
    image: 'dodasts/htcondor:v0.1.0-k8s-schedd-3'
```

Self healing, Volumes management



livenessProbe:

```
exec:
  command:
    - voms-proxy-info
    - '--file'
    - /root/proxy/gwms_proxy
    - '--exists'
    - '--valid'
    - '6:00'
initialDelaySeconds: 300
timeoutSeconds: 1
periodSeconds: 600
successThreshold: 1
failureThreshold: 3
```

volumes:

```
- name: proxydir
  emptyDir: {}
- name: configcondor
  configMap:
    name: condor-configd
    defaultMode: 420
- name: myspool
  persistentVolumeClaim:
    claimName: schedd-claim
```

volumeMounts:

```
- name: myspool
  mountPath: /var/lib/condor/spool/
- name: proxydir
  mountPath: /root/proxy ←
- name: configcondor
  mountPath: /etc/condor/config.d
```


configMap & HTC config



```
[root@vnode-2 config.d]# ls
flockto.10
[root@vnode-2 config.d]# cat flockto.10
FLOCK_TO = 131.154.97.113
FLOCK_COLLECTOR_HOSTS = $(FLOCK_TO)
FLOCK_NEGOTIATOR_HOSTS = $(FLOCK_TO)
HOSTALLOW_NEGOTIATOR_SCHEDD = $(COLLECTOR_HOST), $(FLOCK_NEGOTIATOR_HOSTS)
[root@vnode-2 config.d]#
```

```
volumeMounts:
  - name: myspool
    mountPath: /var/lib/condor/spool/
  - name: proxydir
    mountPath: /root/proxy
  - name: configcondor
    mountPath: /etc/condor/config.d
```

```
1 kind: ConfigMap
2 apiVersion: v1
3 metadata:
4   name: condor-configd
5   namespace: default
6   selfLink: /api/v1/namespaces/default/configmaps/condor-configd
7   uid: 843ad244-f59c-4025-835e-168ae88b3b3b
8   resourceVersion: '41736419'
9   creationTimestamp: '2020-06-17T06:47:23Z'
10 data:
11   flockto.10: |
12     FLOCK_TO = 131.154.97.113
13     FLOCK_COLLECTOR_HOSTS = $(FLOCK_TO)
14     FLOCK_NEGOTIATOR_HOSTS = $(FLOCK_TO)
15     HOSTALLOW_NEGOTIATOR_SCHEDD = $(COLLECTOR_HOST), $(FLOCK_NEGOTIATOR_HOSTS)
```

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: wnconfigd
  namespace: default
  selfLink: /api/v1/namespaces/default/configmaps/wncon
  uid: c975791a-f6fb-4cd9-8345-50ef31a4da27
  resourceVersion: '48642116'
  creationTimestamp: '2020-06-24T13:29:43Z'
data:
  01_DODAS_Custom: |
    Group = "Fermi"
    STARTD_ATTRS = $(STARTD_ATTRS) Group
    START = ( $(START) ) && (TARGET.Group == "Fermi")
```


Spool directory



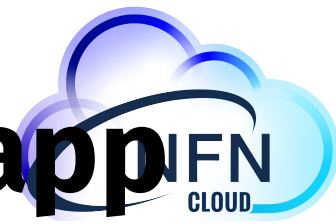
Persistent Volume

- The spool is there at any time k8s restarts the schedd service

```
volumeMounts:
  - name: myspool
    mountPath: /var/lib/condor/spool/
  - name: proxydir
    mountPath: /root/proxy
  - name: configcondor
    mountPath: /etc/condor/config.d
```

```
[root@vnode-2 ~]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
overlay         20263528  7940076  12307068  40% /
tmpfs           65536        0     65536    0% /dev
tmpfs          2023100        0    2023100  0% /sys/fs/cgroup
/dev/vda1       20263528  7940076  12307068  40% /app
shm             65536        0     65536    0% /dev/shm
/dev/vdc1       287830400  108956  273077476  1% /var/lib/condor/spool
```

Finally a customization example: flask app

A screenshot of a web browser window showing a registration form. The address bar displays '193.204.89.89:48080/register'. The form has two input fields: 'Username' and 'IAM-Access-Token', and a 'Register' button below them. Arrows from the browser window point to the code blocks on the right and bottom.

```
40     try:
41         DN = err.split("UserDN: ")[1].replace("/", "\\").rstrip()
42     except Exception as ex:
43         logging.error("failed to get dn from: %s",
44                       form.username.data, ex)
45         return render_template('register.html', DN, form=form)
46
47     with open('/home/uwdir/condormapfile', 'r') as condor_file:
48         old = condor_file.read()
49         with open('/home/uwdir/temp_file', 'w') as temp_file:
50             entry = "GSI \"^\" + DN + \"$\" " + form.username.data + " \n"
51             temp_file.write(entry)
52             temp_file.write(old)
53         os.rename('/home/uwdir/temp_file', '/home/uwdir/condormapfile')
```

```
[root@vnode-2 ~]# cat /home/uwdir/condormapfile
GSI "^\\C=IT\\O=CLOUD@CNAF\\CN=1e7074e5-96fe-43e8-881d-4d572c128931@dodas-iam$" dciangot
GSI "^\\C=IT\\O=CLOUD@CNAF\\CN=6117ac96-08fd-418c-82f5-9eddd57c6b04@dodas-iam$" rangioni
GSI "^\\C=IT\\O=CLOUD@CNAF\\CN=ea88f310-1af1-4bbd-91a2-45dbddaa6445@dodas-iam$" duranti
GSI "^\\C=IT\\O=CLOUD@CNAF\\CN=7568dc96-e218-4d63-a616-ec3ba3956df6@dodas-iam$" vformato
GSI "^\\C=IT\\O=CLOUD@CNAF\\CN=1e7074e5-96fe-43e8-881d-4d572c128931@dodas-iam$" dciangot
GSI "^\\C=IT\\O=CLOUD@CNAF\\CN=0931b26e-89f6-4118-a5c4-dd7f9e9ec85a@dodas-iam$" spiga
```



HTCondor and Docker: Official repo

Nowadays officially supported by HTConor project:

<https://github.com/htcondor/htcondor/tree/master/build/docker/services>

- Execute Node (htcondor/execute)
- Central Manager (htcondor/cm)
- Submit Node (htcondor/submit)
- Minicondor (htcondor/mini)

How to give it a try:

```
dockerhost$ docker run --detach --name=minicondor htcondor/mini:e17
```

Dockerfile: the submit node example



26 lines (19 sloc) | 661 Bytes

```
1 # This is a submit host image for HTCondor with a single user for submission
2 ARG EL
3 ARG VERSION
4 ARG SUFFIX
5
6 FROM htcondor/base:${VERSION}-el${EL}${SUFFIX}
7
8 ARG EL
9 ARG SERIES
10 ARG BUILDDATE
11 ARG SUFFIX
12
13 # https://label-schema.org/rc1
14 LABEL org.label-schema.name="htcondor/submit:${VERSION}-el${EL}${SUFFIX}" \
15       org.label-schema.description="HTCondor ${VERSION} submit host image for RH
16       org.label-schema.vendor="HTCondor" \
17       org.label-schema.license="Apache-2.0"
18
19 # Add a test submitter user
20 RUN useradd submituser
21
22 COPY submit/condor/*.conf /etc/condor/config.d/
23
24 EXPOSE 9618
25
26 LABEL org.label-schema.build-date="${BUILDDATE}"
```

```
26 RUN \
27     # Get the release series based on the middle version
28     # odd numbers = development; even numbers = stable
29     tmp=${VERSION%.*}; \
30     tmp=${tmp#*}; \
31     if (( (tmp % 2) == 1 )); then \
32         SERIES=development; \
33     else \
34         SERIES=stable; \
35     fi; \
36     \
37     yum -y update && \
38     yum -y install epel-release yum-plugin-priorities && \
39     \
40     (curl -sSL https://research.cs.wisc.edu/htcondor/yum/repos.d/htcondor-${SERIES}-rhel${EL}.repo && \
41     echo "ggpkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-HTCondor" \
42     ) > /etc/yum.repos.d/htcondor-${SERIES}-rhel${EL}.repo && \
43     curl -sSL https://research.cs.wisc.edu/htcondor/yum/RPM-GPG-KEY-HTCondor \
44     -o /etc/pki/rpm-gpg/RPM-GPG-KEY-HTCondor && \
45     rpm --import /etc/pki/rpm-gpg/* && \
46     \
47     yum -y install "condor = ${VERSION}" supervisor openssh-clients openssh-server && \
48     yum clean all && \
49     rm -rf /var/cache/yum/*
50
```

1 contributor

1 lines (1 sloc) | 18 Bytes

1 use ROLE : Submit



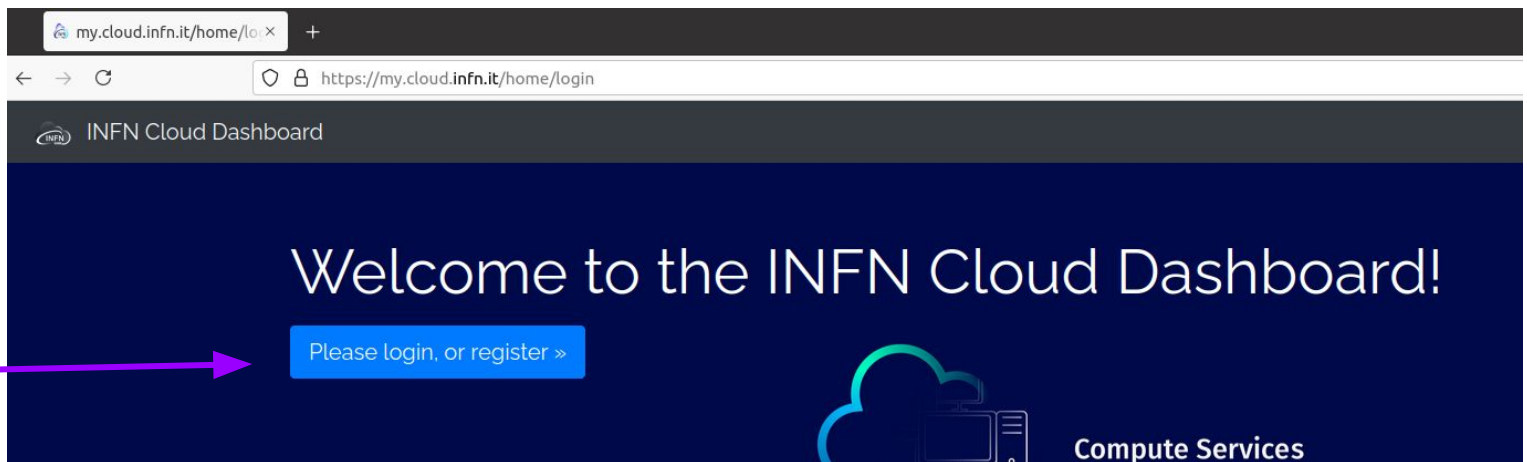
And finally to summarize...

INFN-Cloud in practice

All INFN staff and associates can exploit INFN-Cloud resources

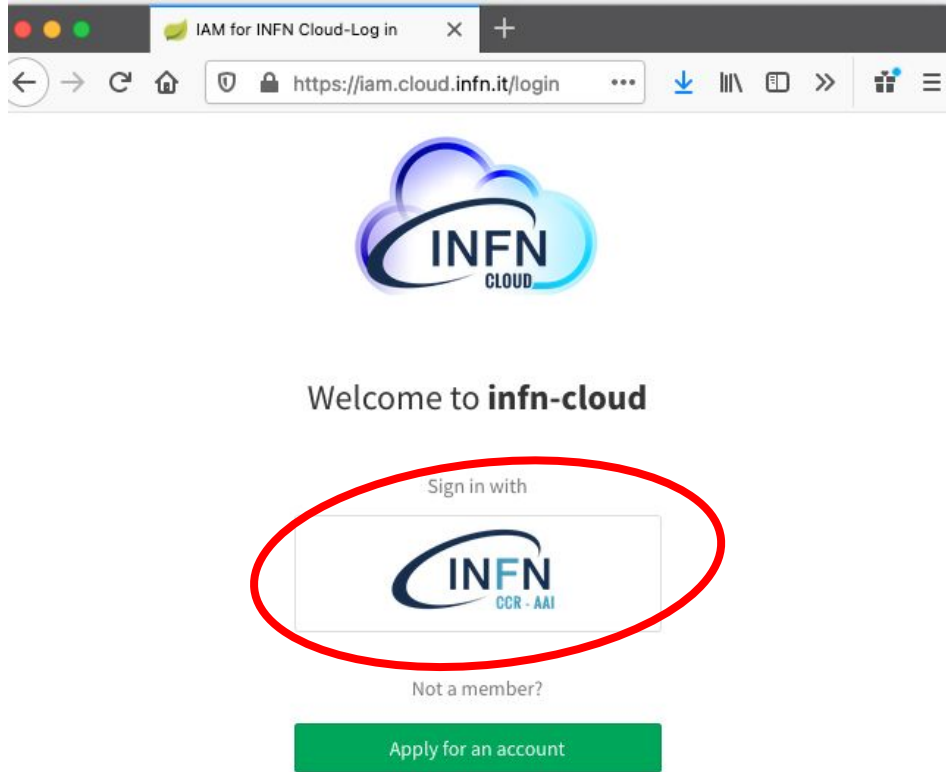
- This means storage and compute services. Compute means not only simple VMs but also composed services (such as the Jupyter-based one used by ML_INFN). **You can also customize and personalize your own environment [see later]**
- **By default, a INFN-Cloud user has a fixed and limited quota (i.e. #Cores/RAM)**
 - Experiments/collaborations follow a different path!

If you are interested either to continue working as you did during the hackathon or even to do something else visit [here](#)



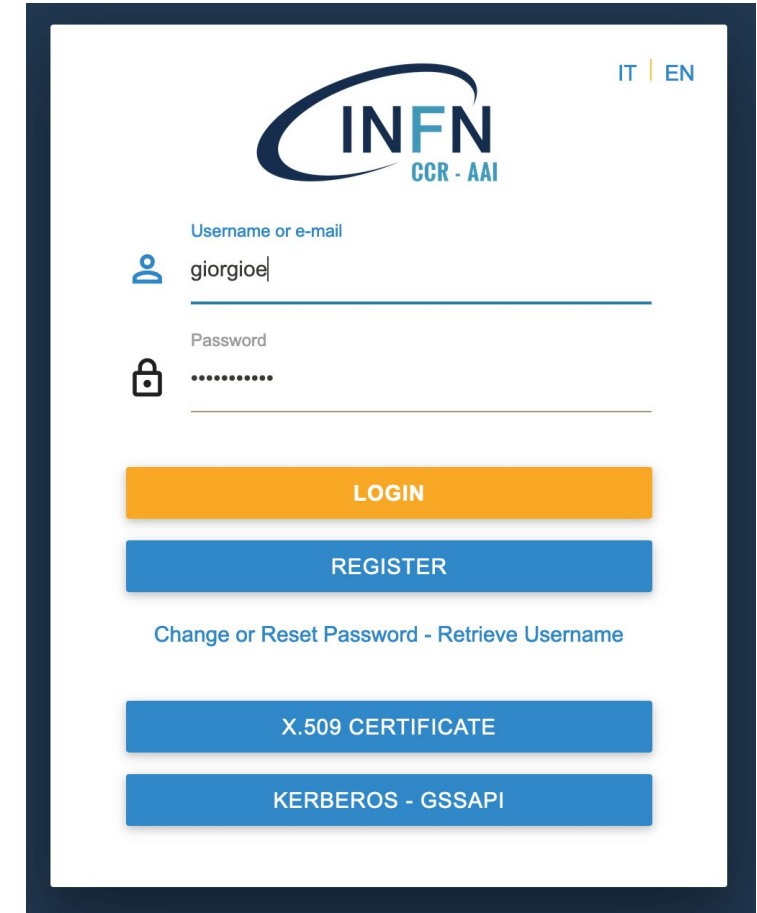


Once you click you will be redirected to the IAM account registration



Don't "Apply for an account" but

Sign in with your own INFN AAI identity



NOTE: Once done you'll be contacted by INFN-Cloud support team

IMPORTANT

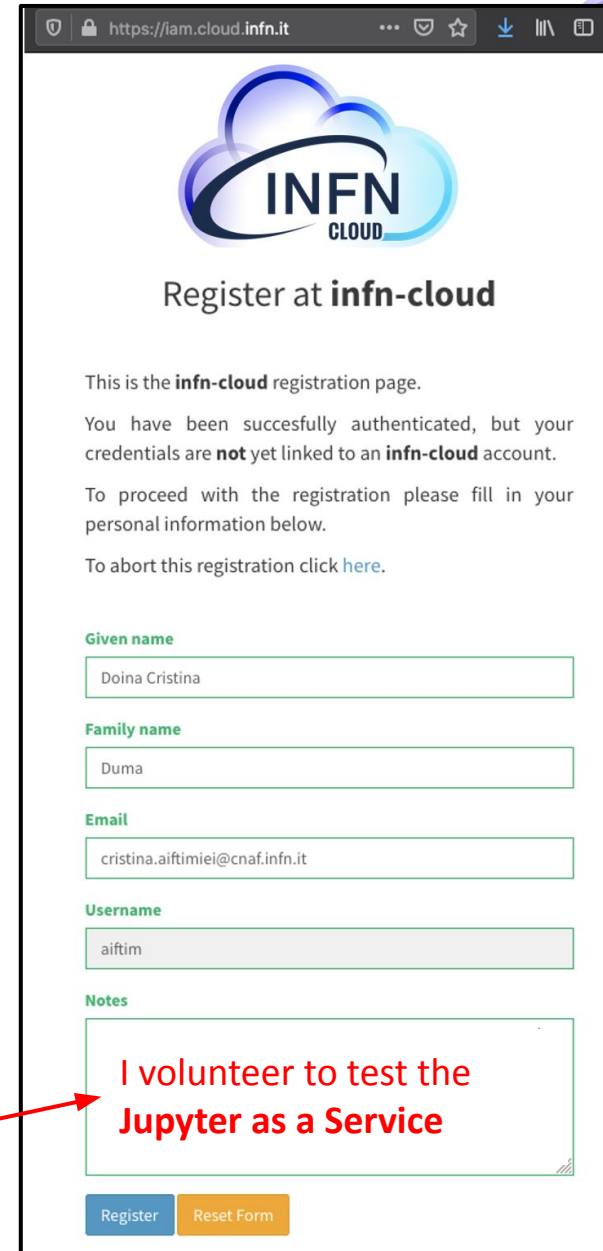
There are two pre-requisites

- **Digital identity** on INFN AAI and **acceptance of INFN usage rules for IT resources**
 - INFN staff or associates meet these requirements by default
 - Others see <https://signup.app.infn.it/>
- **Designation** as 'System administrator' for INFN Cloud granted by your own INFN Director.
 - Consult your local INFN section or laboratory for further details

If you don't want/can't get the Designation, we are working for you :).. We are releasing a Jupyter as a Service instance that can be used even without Designation!

- When you register, just be sure you fill the note with this and we will contact you

INFN Cloud



https://iam.cloud.infn.it

INFN
CLOUD

Register at **infn-cloud**

This is the **infn-cloud** registration page.

You have been successfully authenticated, but your credentials are **not** yet linked to an **infn-cloud** account.

To proceed with the registration please fill in your personal information below.

To abort this registration click [here](#).

Given name

Family name

Email

Username

Notes

Register Reset Form



And for those who wants to raise the bar

INFN-Cloud allows you to implement new services and/or customize existing one

Examples:

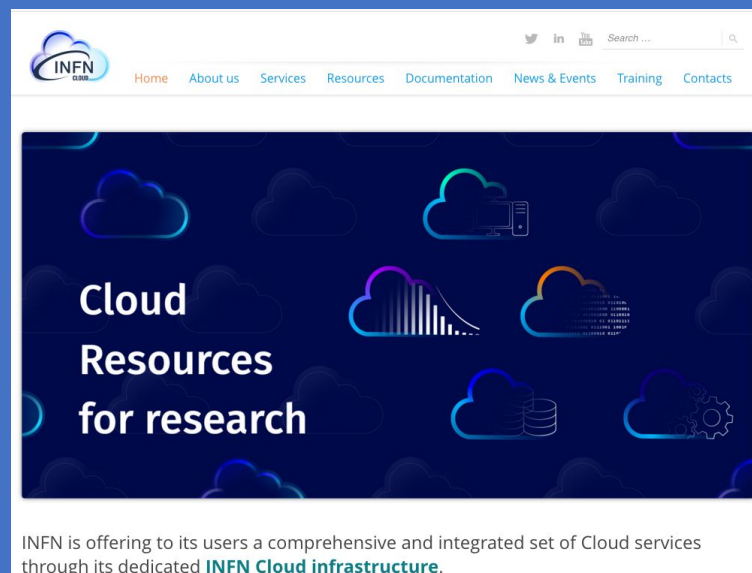
- “I want to build MY Jupyter-based workflow and possibly share it within my collaboration”
- “ I would like to use INFN Cloud but I don’t need/like Jupyter, I would rather like to run my small cluster ...”

In such a case please contact us at: cloud-support@infn.it and you will be redirected to the proper INFN Cloud support team.



Reference

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



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



Support :
<https://servicedesk.cloud.infn.it> or  cloud-support@infn.it

