

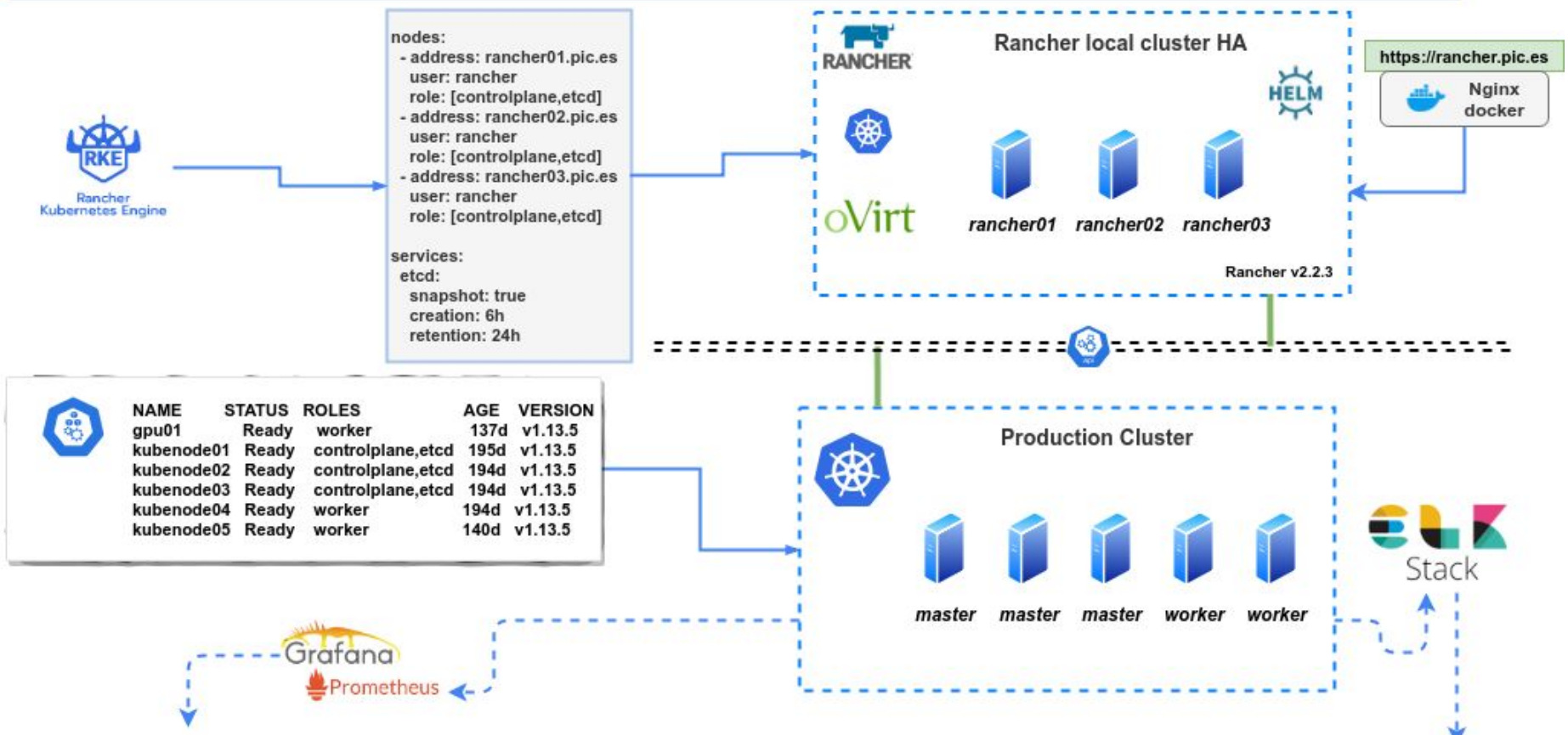
# k8s at PIC

current and short term deployments

Vanessa Acín, Ricard Cruz, Carles Acosta, Bruno Rodriguez

- **Two k8s platforms deployed with Rancher:**
  - Production cluster (rancher v-2.2.3., k8s v-1.13.5)
    - Resources:
      - Rancher cluster: 3xVM
      - k8s cluster: 3xVM for the master nodes, 2xVM workers, 1 GPU physical server as a worker
    - Services: JupyterHUB and HTCondor GPU-WN
    - Users: local ATLAS and MAGIC users and Grid Computing resources for VIRGO
  - Test cluster (rancher v-2.2.8., k8s v-1.15.3)
    - Future services: Xcache and frontier-squid instances
    - Users: VIRGO and ALTAS

# Kubernetes Architecture with RANCHER



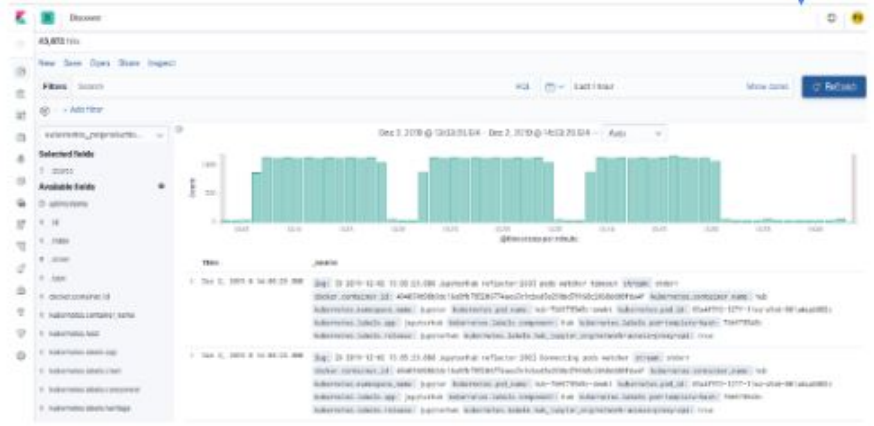
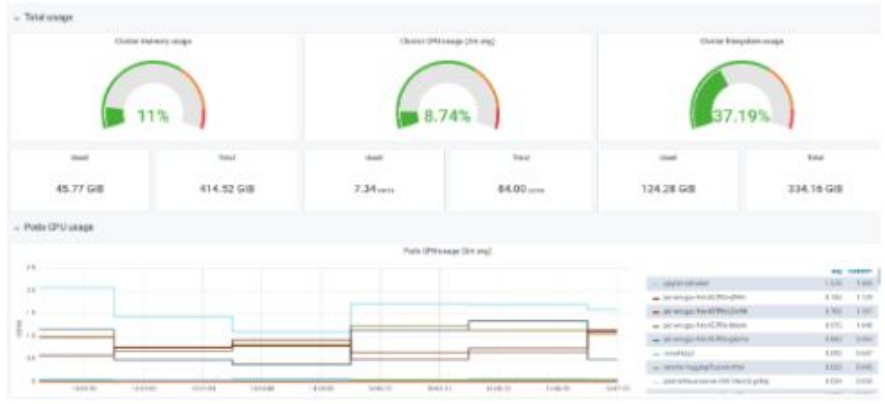
**nodes:**

- address: rancher01.pic.es  
user: rancher  
role: [controlplane,etcd]
- address: rancher02.pic.es  
user: rancher  
role: [controlplane,etcd]
- address: rancher03.pic.es  
user: rancher  
role: [controlplane,etcd]

**services:**

etcd:  
snapshot: true  
creation: 6h  
retention: 24h

NAME	STATUS	ROLES	AGE	VERSION
gpu01	Ready	worker	137d	v1.13.5
kubenode01	Ready	controlplane,etcd	195d	v1.13.5
kubenode02	Ready	controlplane,etcd	194d	v1.13.5
kubenode03	Ready	controlplane,etcd	194d	v1.13.5
kubenode04	Ready	worker	194d	v1.13.5
kubenode05	Ready	worker	140d	v1.13.5



One 8xGPUs physical server sharing resources between two services

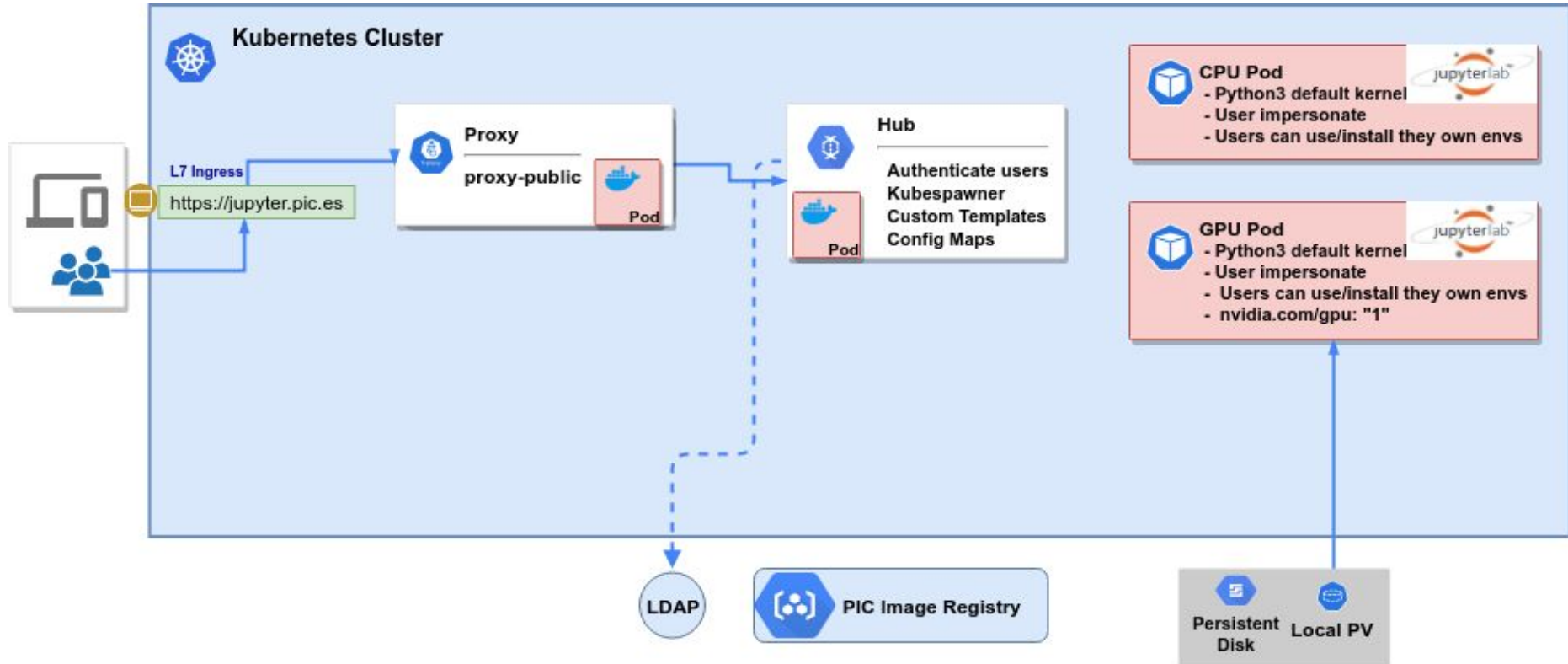
- **Jupyterhub**

- Interactive python notebook service only for local users
- Two different profiles: CPU and GPU pod
- LDAP user authentication
- Access to local \$HOME disk area space and an scratch directory
- Default Conda environment mounted via NFS to the pod
  - Python 3
  - JupyterLab 1.0.0
  - Notebook 6.0.0
- Users can use their own environment

- **HTCondor GPU workernodes**

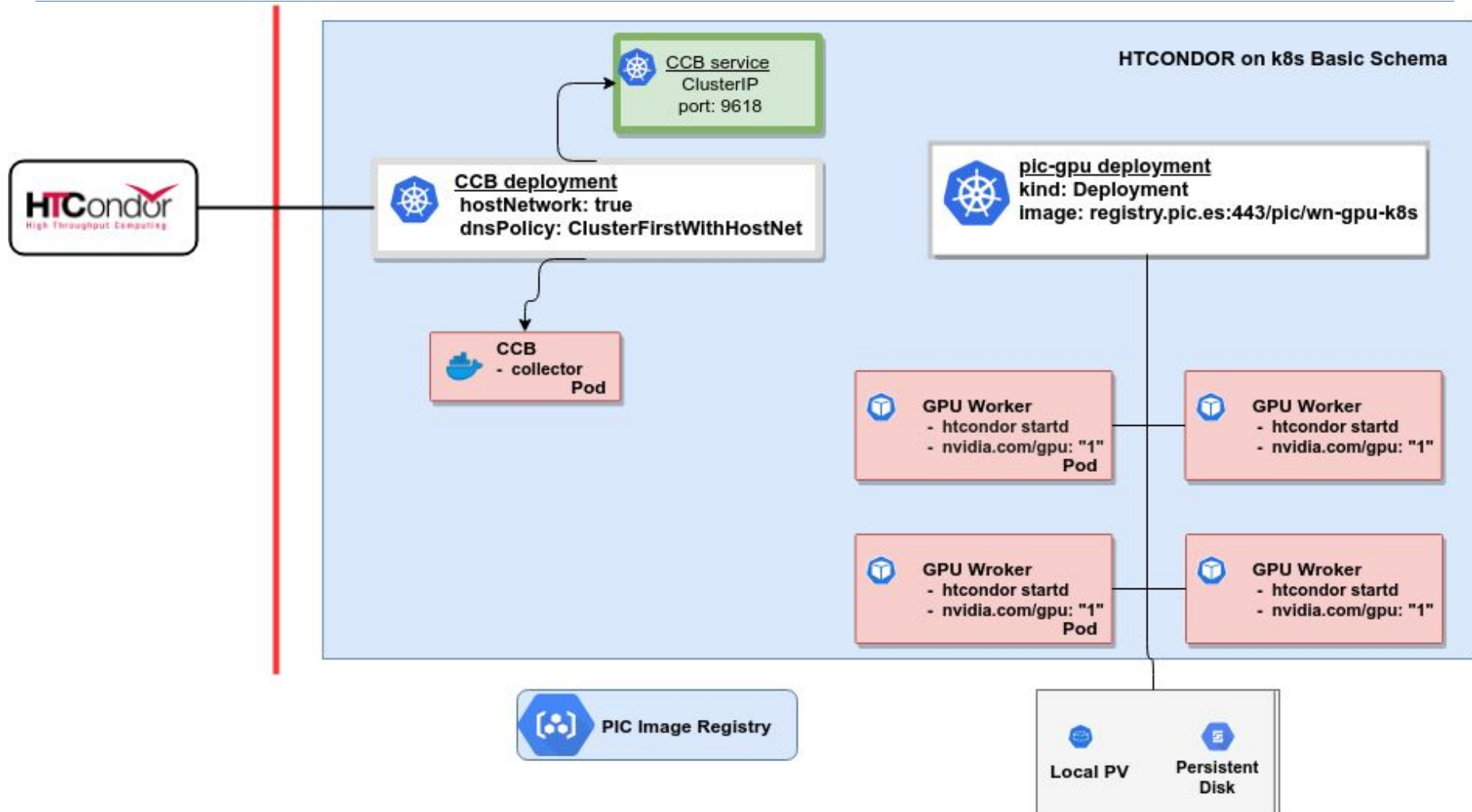
- Two different k8s deployments: CCB and workernode startd
- HTCondor Schedd accesses the workernodes through the CCB

# Jupyterhub diagram



Based on Zero to JupyterHUB: <https://zero-to-jupyterhub.readthedocs.io/en/latest/>

# HTCondor GPU diagram



Based on OSG HTCondor for PRP <https://github.com/sfiligoi/prp-osg-pool>

- **Improve redundancy**
  - add more GPU resources
  - need to improve redundancy for proxy and hub services
- **Operation**
  - user accounting for jupyter resources
  - improve k8s cluster monitoring
  - team training
- **Resource sharing**
  - Flexible GPU assignment between services
  - Prioritize user interactive demand versus batch computing ( using PriorityClass and preemption with k8s)

Main goal: get hands-on experience

- **Xcache for VIRGO**
  - The experiment provides the deployment pattern
- **Frontier-squid instances**
  - One k8s Service for each instance
  - CephFS as cache volume
- **Upgrade production cluster**
- **Move all deployments in the same cluster**



Thank you for your attention