Part 1.4: TOSCA Orchestration and HPCWaaS

Jorge Ejarque (BSC)

ISC-HPC 23 – Hamburg 21st of May 2023

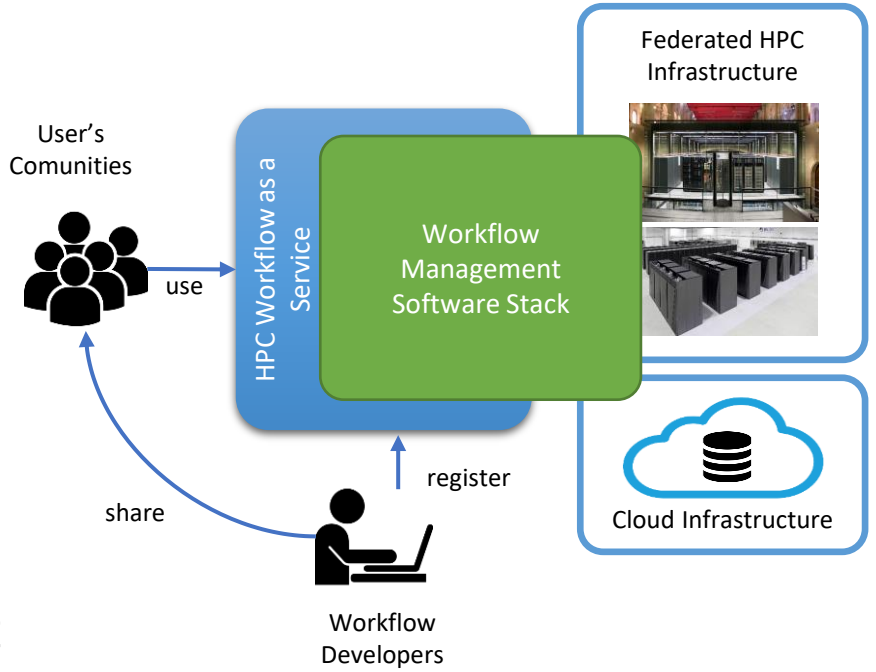
This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway. MCIN/AEI/10.13039/501100011033 and the European Union NextGenerationEU/PRTR (PCI2021-121957)

Recap

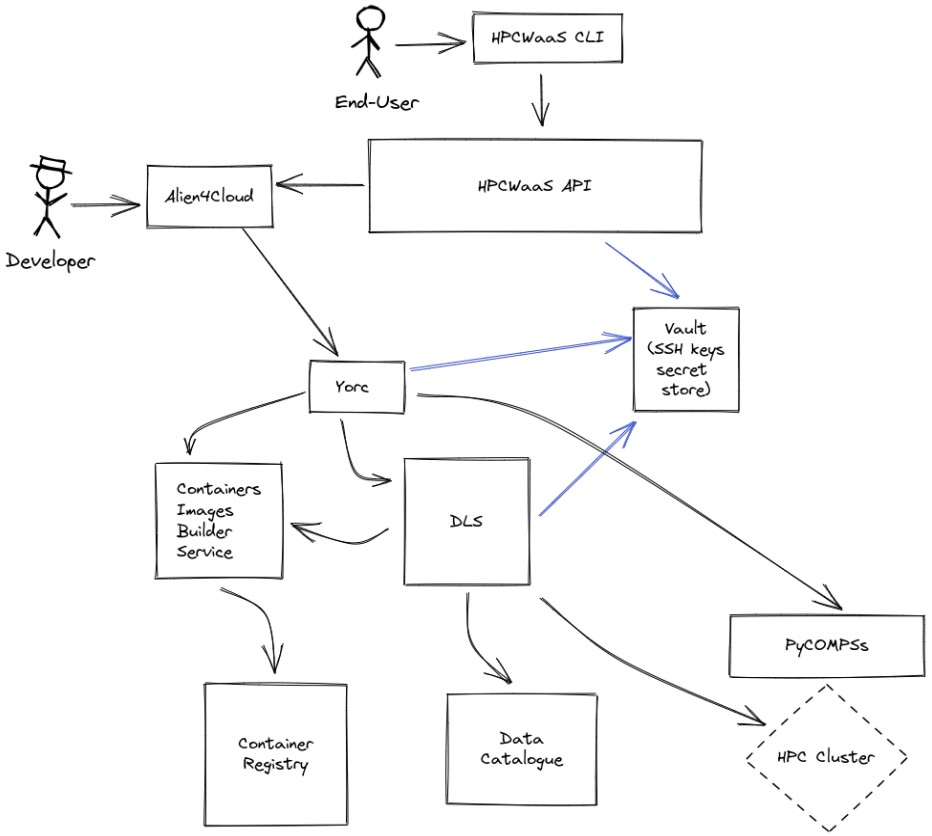
- Part 1: Implementing computation
- Part 2: Create containers
- Part 3: Dealing with data logistics
- Let’s see how to orchestrate all together

eflows4HPC approach

- **Require a description for the workflow lifecycle management**
  - TOSCA:
    - Model to describe cloud application topologies and the lifecycle orchestration
- **Interface for deploying and running the workflows**
  - HPCWaaS:
    - Development and Deployment (Alien4Cloud)
    - Execution (HPCWaaS API)



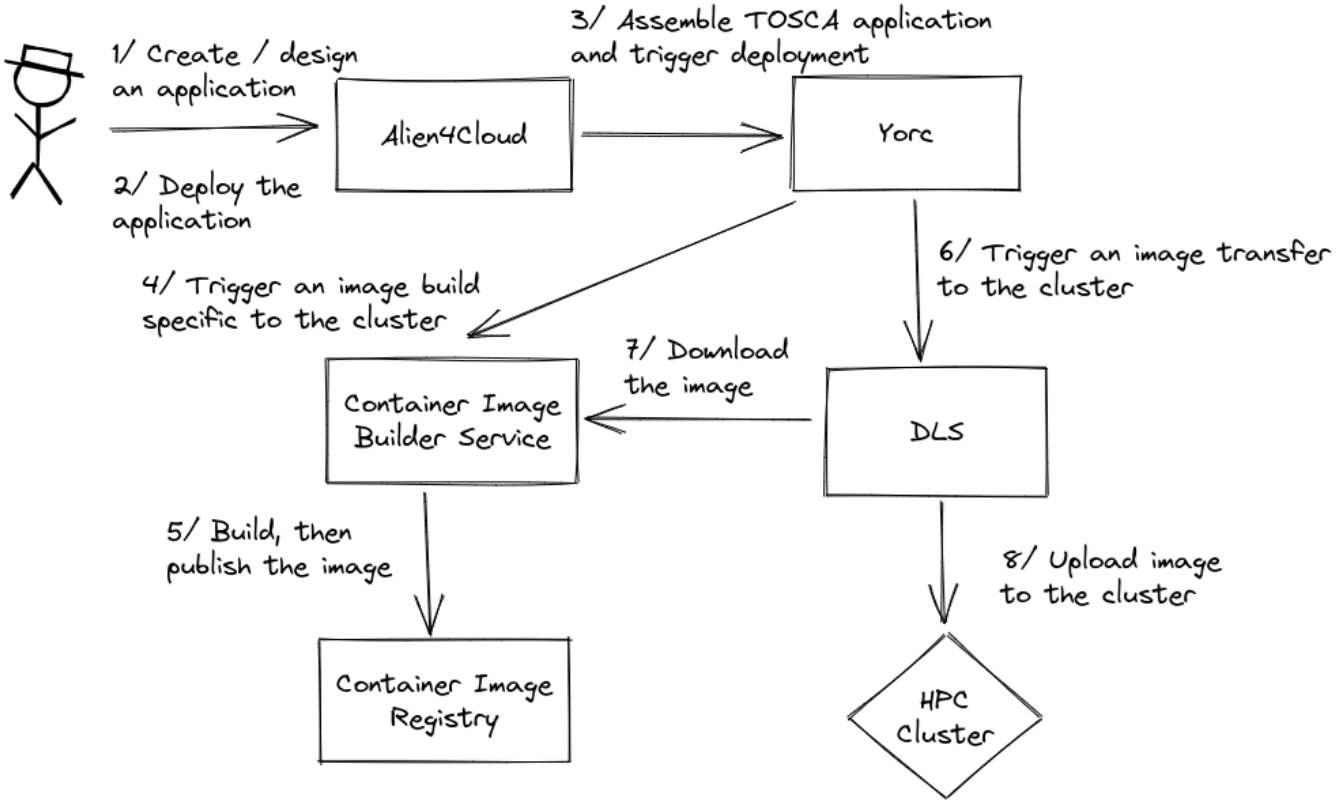
Overview



- HPCWaaS API:
  - control workflow executions
- Alien4Cloud:
  - design & deploy workflows
- Yorc:
  - high level orchestration engine
- Vault:
  - securely store credentials
- Container Image Builder service:
  - build container images
- DLS:
  - data movements engine
- PyCOMPSs:
  - Runtime engine for computations
- Container registry:
  - store container images
- Data Catalogue:
  - Registry for datasets locations

DEVELOPER POINT OF VIEW

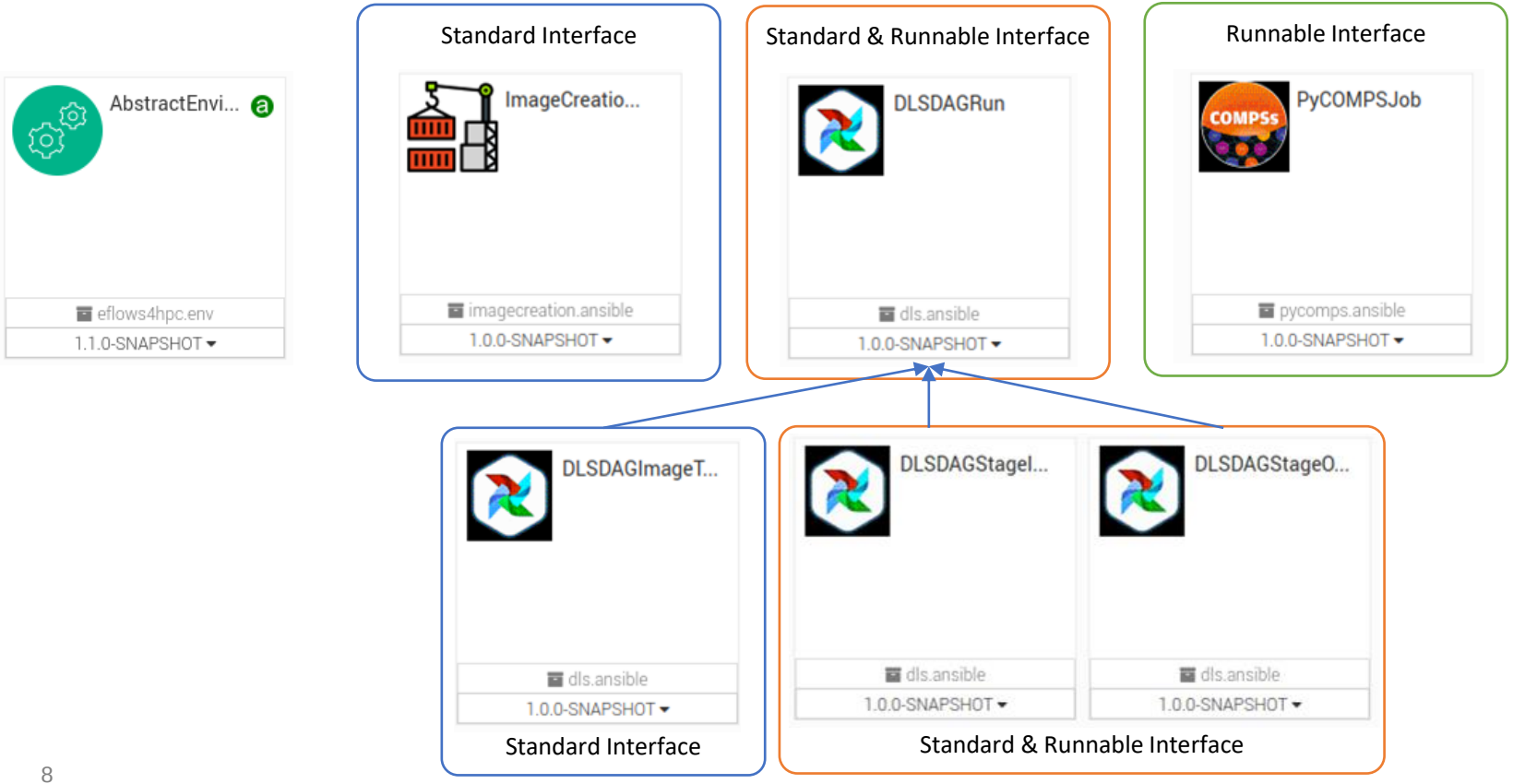
Workflows development and deployment



TOSCA Model

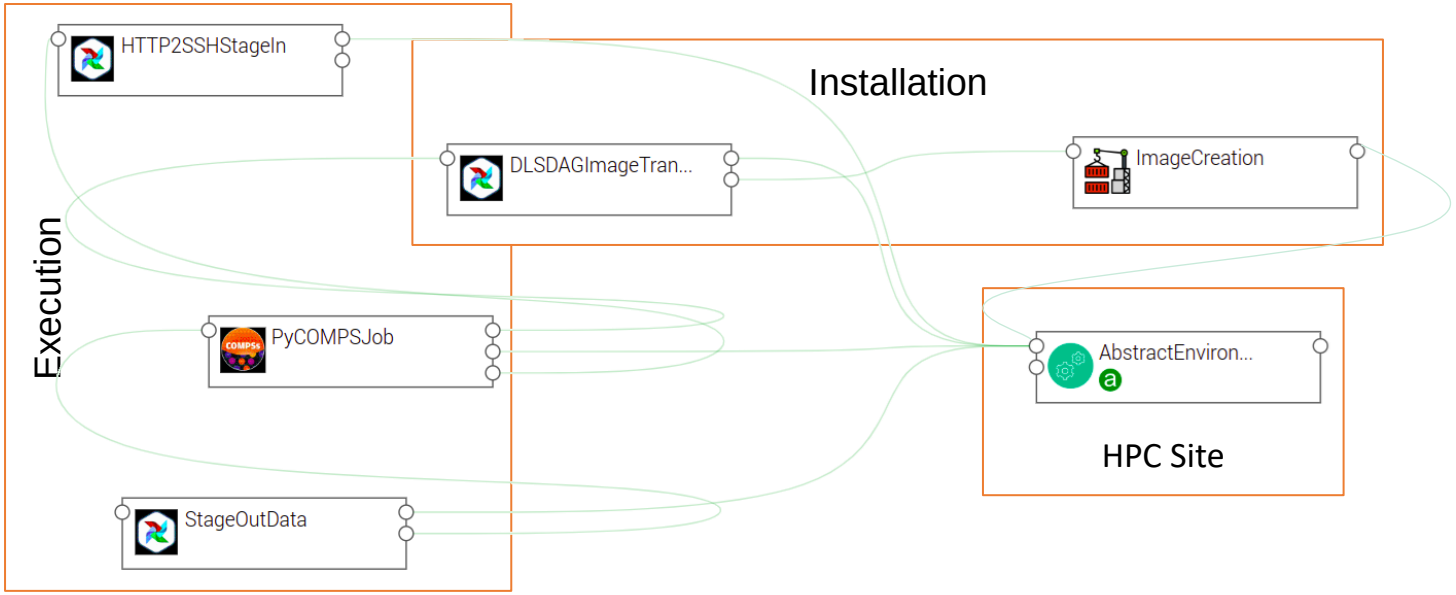
- **Describe the orchestration of the application lifecycle management**
- **Topology of components with dependencies**
  - Application Component:
    - Describe what to do in every lifecycle step
      - ✓ Standard tosca steps (start, stop, delete,…)
      - ✓ Extended runnable (submit, run, cancel,…) Integrate jobs in Tosca.
    - The required input data and properties
  - Dependencies:
    - Describe the data exchanged between components.
- **Workflows**
  - Topology generate the standard TOSCA workflows to deploy/undeploy the application
  - Custom workflows

eFlows4HPC TOSCA Components



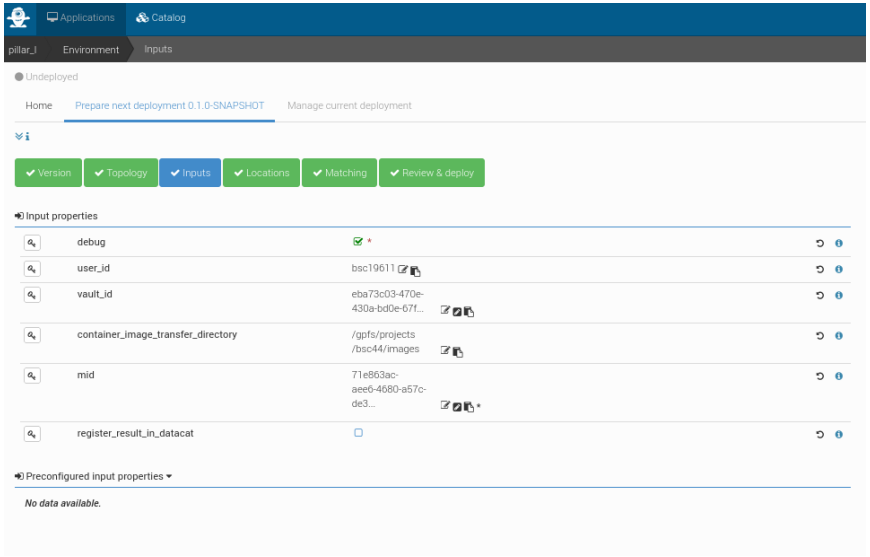
TOSCA Modelization

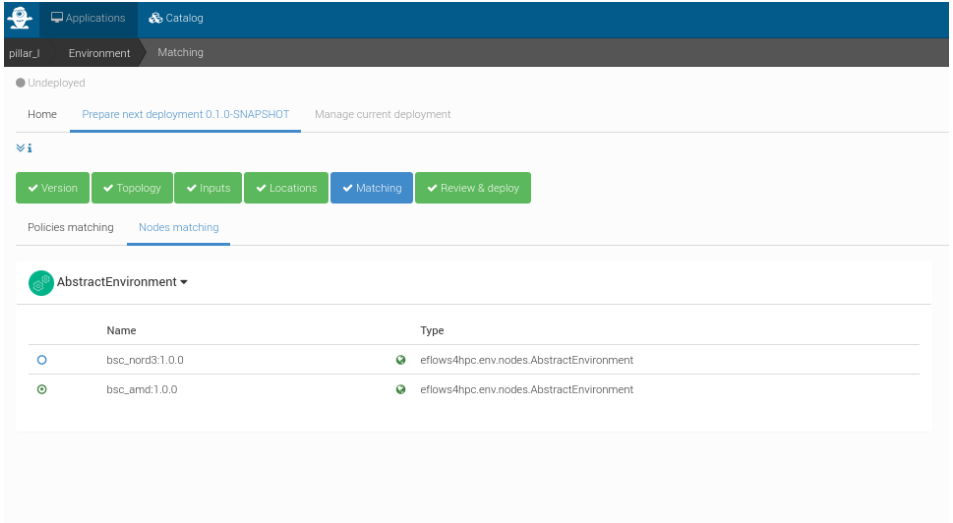
Topology of the different components involved in the Workflow lifecycle

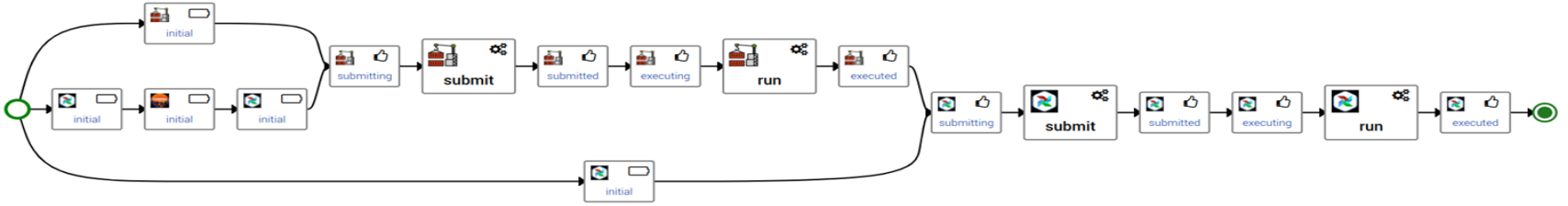


Workflow Deployment (done once per HPC site)

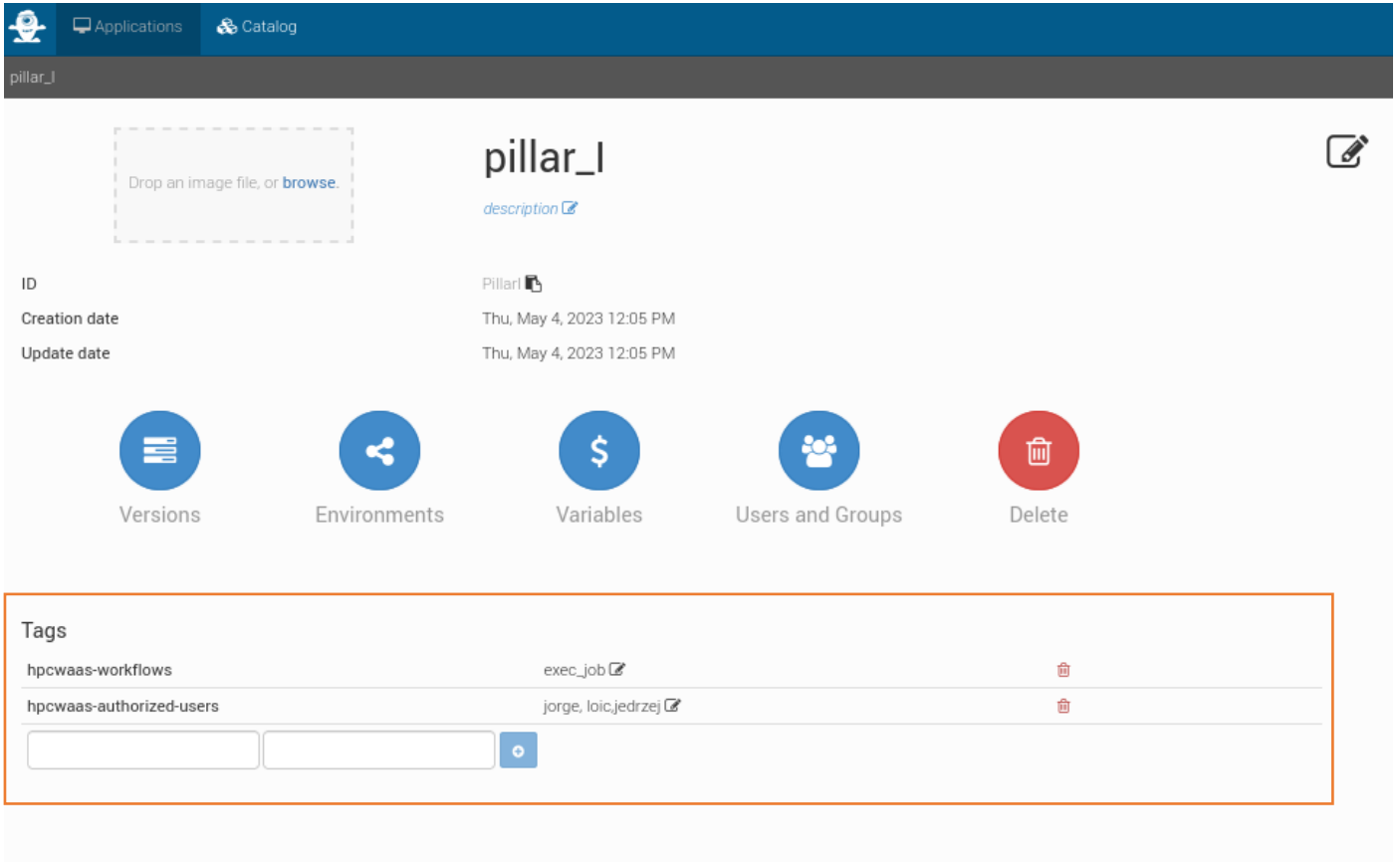
- Set deployment input parameters (user, credential, select HPC location)





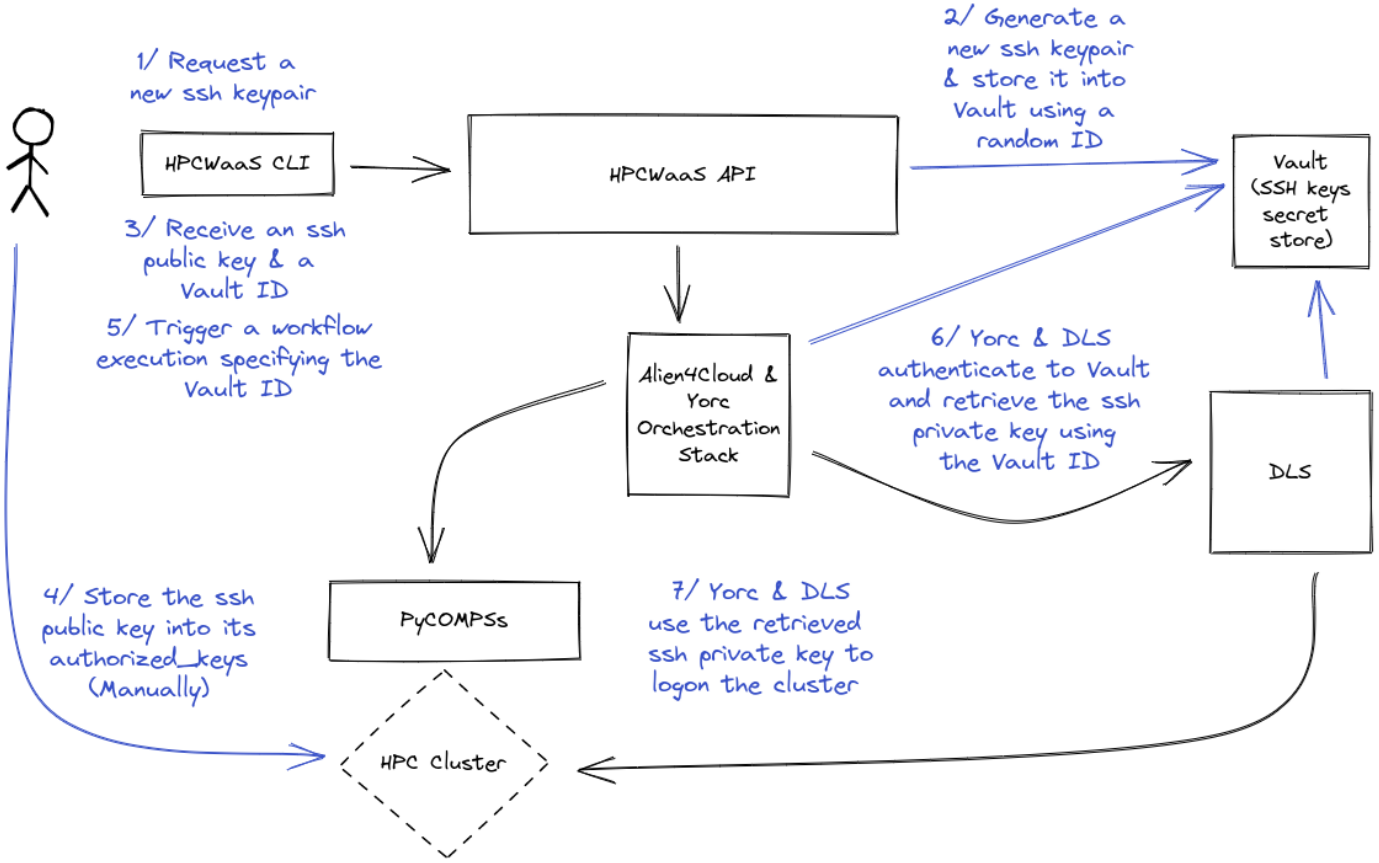


Publish workflow and authorize users



END-USER POINT OF VIEW

Workflow Execution End user



HPCWaaS main CLI commands

- **waas ssh_keys key-gen**
  - Generate SSH keys to access the HPC on behalf of the user
- **waas workflows list**
  - List workflows accessible by the user
- **waas workflows trigger**
  - Lauch the execution of a workflow
- **waas executions status**
  - Check the status of the execution
- **waas executions cancel**
  - Cancel a workflow execution

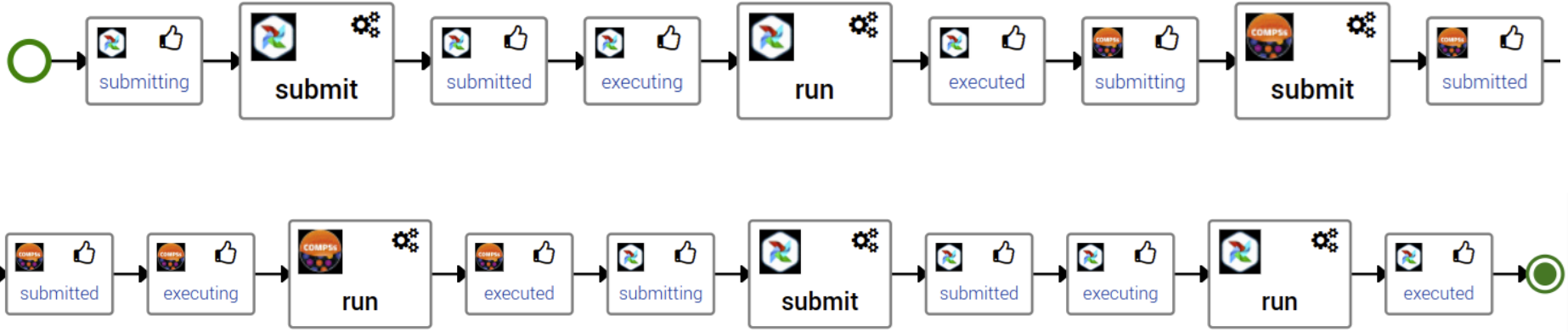
SSH Key Generation

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> ssh_keys key-gen
INFO: Below is your newly generated SSH public key.
INFO: Take note of it as you will not see it again.
INFO: You are responsible for adding it to the authorized_keys file on the systems you want to run your workflows.
INFO: SSH key ID:     31…3f
INFO: SSH Public key: ssh-rsa AAA…mH
```

- SSH Key ID submitted to the workflow execution
- SSH Public Key must be added by the user in their authorized keys

Workflow Execution

```
./waas --api_url https://eflows4hpc.bsc.es/waas -u <user>:<password> workflows
trigger -f -i user_id=<username> -i vault_id=<SSH_KEYPAIR_ID> -i
oid=2c2463377aac4aa59381c6b06fe800f3 -i
target_path=/home/nct01/<username>/data/inputs -i
source_path=/home/nct01/<username>/data/results -i num_nodes=2 <workflow_id>
```



Conclusion

- **HPC Workflow as a Service**
  - Alien4Cloud: Describe and deploy HPC workflow as a TOSCA Application
  - Execution API: Manage SSH credentials and execution of deployed workflows

eFlows4HPC

Enabling dynamic and Intelligent workflows
in the future EuroHPC ecosystem

www.eFlows4HPC.eu

@eFlows4HPC eFlows4HPC Project

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.