EOSC Technical Specification

Hub Federated Core Services

HTC/HPC Compute - HPC/HTC clusters on demand

Introduction

Scientific portals expose convenient interfaces that typically run partially-customisable jobs on computing infrastructures. Scientific portals normally have a higher computational demand on their back-ends than conventional portals, so they have to be provisioned with enough resources to deal with the potentially unexpected workload peaks. However, users in scientific portals could also allow longer delays on retrieving the results, as they are more used to queuing systems.

In this regard, we identify the need for provisioning self-managed elastic clusters supporting mainstream job managers such as PBS, SLURM and especially scheduling system based on Kubernetes resource orchestrators. Opposite to the multitenant job management service, this service will explicitly deploy a single-tenant cluster backend to be used by the user community managed by the user who deployed it.

Adopted Standards

Standard	Short description	References
TOSCA	OASIS Topology and Orchestration Specification for Cloud Applications, a specification of infrastructure and	https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

	applications as code.	
OpenStack (especially Nova API and Keystone API v3)	OpenStack is an Open Source cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.	OpenStack API
OpenID tokens & SAML	The service must leverage habitual authentication and authorization mechanisms used in Scientific Research Infrastructures to avoid users to handle multiple credentials. EGI Check-in is an example (https://wiki.egi.eu/wiki/AAI_guide_for_SPs)	https://openid.net/connect/, http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.pdf

High-level Service Architecture

This service can be built as an added value service on top of the cloud orchestration services. A user should be able to easily deploy a cluster through a service that offers some customisation options, such as the Local Resource Management System (LRMS), the flavour (the basic hardware configuration of a cloud instance, such as the number of virtual cpus, the memory size, the access to specific physical devices) and virtual machine image (a snapshot or virtual disk that contains the operative system and the software dependencies) for the front end and working nodes, the maximum number of nodes that the cluster can grow up, additional software dependencies and even the support of hybrid deployments with different working node types.

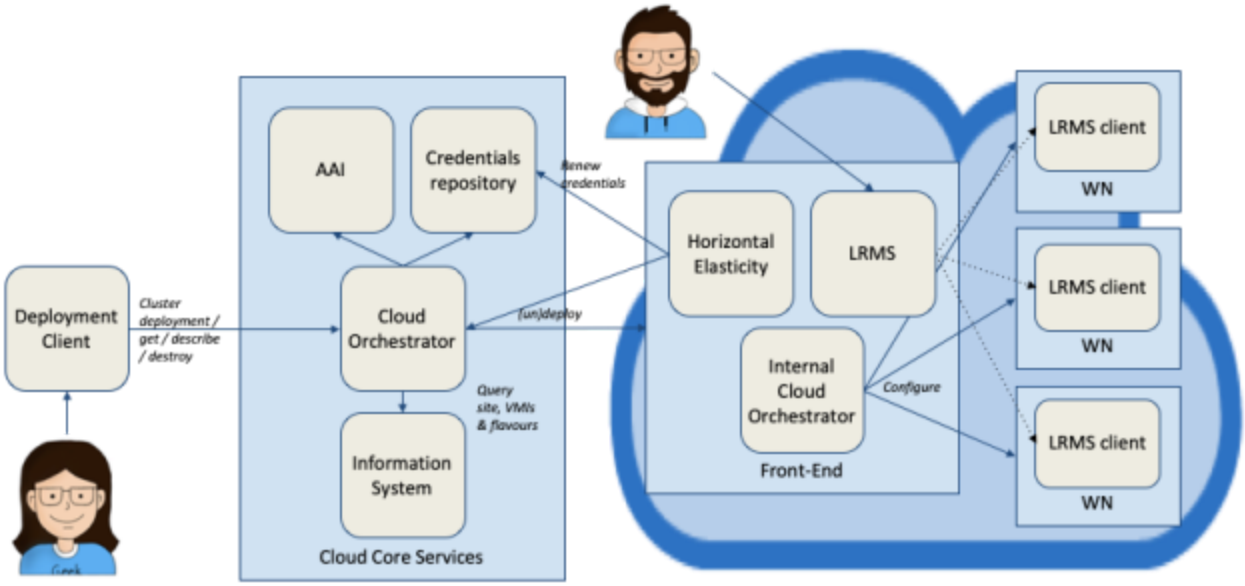

Figura 1: Potential high-level architecture for the Clusters on-demand macro feature.

The components described in figure 1 are:

- Deployment client. An application that takes a cluster specification from the user and interacts with a cloud orchestrator for its deployment and configuration. The Deployment client should reduce the complexity of building the configuration specification and the interaction with the cloud orchestrator but also provide a high level of flexibility for advanced users.
- Internal Cloud Orchestrator. The start up of a cluster will require interacting with IaaS sites for deploying the resources, potentially installing software and configuring them. We differentiate from an external Cloud Orchestrator Service from the Internal service as a suggestion of implementation that has been used in Elastic Compute Clusters in the Cloud to isolate the traffic during the reconfiguration of the nodes and to avoid keeping resource access credentials on the central service.
- The Horizontal Elasticity module that will query the LRMS and interact with the Internal Cloud Orchestrator to update the cluster configuration.
- Cloud core services. This specification will come from other building block documents, but in principle, it should comprise a cloud orchestrator that deals with the IaaS, potentially deal with different sites as back-end (an information system should provide the details of the availability of flavours, virtual machine images and resource availability), and should interact with the AAI (Authentication and Authorisation Infrastructure), enabling credential delegation for managing the horizontal elasticity.
- Cloud cluster itself. It will depend on the specific technology of the Local Resource Management System (LRMS). In order to implement horizontal elasticity, modified versions of the LRMS may be requested to retrieve the information from the queue and to update the Working Nodes registered to the queue. Batch queue systems such as SLURM and Torque and Container management systems such as Kubernetes will be the preferred options.

Interoperability guidelines

With respect to the user, the service should expose the following interface:

- Deployment: An interface to customise the cluster according to the features defined in the previous section, ending up with the credentials to access such cluster.
- Management: An interface to browse existing clusters and to delete them.

It should interact with other services of the system, in the following way:

AAI interoperability

- Services should provide access to users authenticated with one of the EOSC-hub AAI federated identity protocols (OpenID Connect and/or SAML).
- The capability of store short-living credentials for managing elasticity, by storing proxies or delegated credentials that can be revoked. This will require interaction with credential store services to refresh them.

Orchestration interoperability

- Services will make use of the API of the cloud orchestration services to deal with the cloud backends.

Examples of solutions implementing this specification

This solution is partially addressed by the following technologies:

- EC3: Elastic Compute Clusters in the Cloud. https://servproject.i3m.upv.es/ec3/
- EKaaS: Elastic Kubernetes Clusters as a Service. https://github.com/grycap/ec3-web/tree/ltos and https://docs.google.com/document/d/10PIS5wwpwQgps6Kxsr7jxvInRLI6G2DXwCvDBWrlPs8/edit