Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Implementation and use of a highly available and innovative IaaS solution: the Cloud Area Padovana

To cite this article: C Aiftimiei *et al* 2015 *J. Phys.: Conf. Ser.* **664** 022016

Related content

- Monitoring of IaaS and scientific applications on the Cloud using the Elasticsearch ecosystem
  S Bagnasco, D Berzano, A Guarise et al.
- TOSCA-based orchestration of complex clusters at the IaaS level
  M Caballer, G Donvito, G Moltó et al.
- Cloud Environment Automation: from infrastructure deployment to application monitoring
  C. Aiftimiei, A. Costantini, R. Bucchi et al.

Recent citations

- Merging OpenStack-based private clouds: the case of CloudVeneto.it
  Paolo Andreetto *et al*
- The Cloud Area Padovana: from pilot to production
  P Andreetto *et al*

Implementation and use of a highly available and innovative IaaS solution: the Cloud Area Padovana

C Aiftimiei[1 4], P Andreetto[2], S Bertocco[2], M Biasotto[3], S Dal Pra[1], F Costa[2], A Crescente[2], A Dorigo[2], S Fantinel[3], F Fanzago[2], E Frizziero[2], M Gulmini[3], M Michelotto[2], M Sgaravatto[2], S Traldi[2], M Venaruzzo[3], M Verlato[2] and L Zangrando[2]

[1] INFN CNAF, Viale Berti Pichat 6/2, I-40127 Bologna, Italy
[2] INFN, Sezione di Padova, Via Marzolo 8, I-35131 Padova, Italy
[3] INFN, Lab. Naz. di Legnaro, Via Romea 4, 35020 Legnaro, Italy
[4] IFIN-HH, Str. Reactorului 30, Magurele, Romania

E-mail: cloud@lists.pd.infn.it

Abstract. While in the business world the cloud paradigm is typically implemented purchasing resources and services from third party providers (e.g. Amazon), in the scientific environment there's usually the need of on-premises IaaS infrastructures which allow efficient usage of the hardware distributed among (and owned by) different scientific administrative domains. In addition, the requirement of open source adoption has led to the choice of products like OpenStack by many organizations.

We describe a use case of the Italian National Institute for Nuclear Physics (INFN) which resulted in the implementation of a unique cloud service, called 'Cloud Area Padovana', which encompasses resources spread over two different sites: the INFN Legnaro National Laboratories and the INFN Padova division. We describe how this IaaS has been implemented, which technologies have been adopted and how services have been configured in high-availability (HA) mode.

We also discuss how identity and authorization management were implemented, adopting a widely accepted standard architecture based on SAML2 and OpenID: by leveraging the versatility of those standards the integration with authentication federations like IDEM was implemented.

We also discuss some other innovative developments, such as a pluggable scheduler, implemented as an extension of the native OpenStack scheduler, which allows the allocation of resources according to a fair-share based model and which provides a persistent queuing mechanism for handling user requests that can not be immediately served.

Tools, technologies, procedures used to install, configure, monitor, operate this cloud service are also discussed.

Finally we present some examples that show how this IaaS infrastructure is being used.

1. Introduction

While the Grid is being heavily used by big HEP collaborations (e.g. by the LHC experiments), this computing paradigm was often considered not suitable for smaller communities. The limited functionality for interactive access to resources, the authentication and authorization based on X509 certificates and the steep learning curve are some of the issues that explain why the Grid computing model didn't have a widespread adoption.

Small experiments tend to buy their own clusters independently to satisfy their computing needs, and this results in a lot of heterogeneous small sized clusters, often underutilized but insufficient in some periods (e.g. when users are close to deadlines). Besides an overall low efficiency in resource usage, the proliferation of these computing clusters leads to a very high system administration cost.

The cloud model can provide the elasticity required by small experiments at lower costs. The infrastructure is as a service (IaaS): no dedicated machines are reserved, resources and services are activated on demand by users and released when not used.

At the end of 2013, INFN-Padova division and INFN Legnaro National Laboratories (LNL) started a project, called *Cloud Area Padovana*, aimed at providing a cloud-based service for computing and storage resources. This project leverages on the long-standing collaboration between these two INFN sites, located about 10 km from each other. In particular, they have been operating a WLCG Tier-2 Grid facility distributed between the two sites [1], for both ALICE and CMS experiments, for many years.

This paper is organized as follows. Sec. 2 provides an overview of the Cloud Area Padovana. Sec. 3 discusses the network layout, while in sec. 4 we describe how the cloud services have been deployed in high availability mode. Sec. 5 discusses the authentication model chosen in this cloud infrastructure and presents some in-house developments that have been integrated to manage user registration. Sec. 6 discusses another new development: the fair-share scheduler which aims to overcome the default static resource partitioning model, which usually results in a poor efficiency in resource usage. In sec. 7 how this cloud is operated is discussed, while sec. 8 presents some examples of how it is being used. Sec. 9 concludes the article.

2. Overview of the Cloud Area Padovana

The Cloud Area Padovana implements an IaaS (Infrastructure as a Service) cloud.

OpenStack [2] was chosen as middleware framework. Besides being one of the most popular and industry supported open source solutions for clouds, it is widely adopted in the scientific reference domain of INFN. At the time of writing the Havana version of OpenStack is used, but the migration to the IceHouse release is being prepared.

Besides the provision of computing resources (virtual machines that are instantiated by the users on-demand and in self-provisioning mode and released when not needed anymore) a block storage service (implemented through the OpenStack Cinder service) is available. The Swift OpenStack object storage service was instead not deployed, since no use-cases for its possible use were identified. Cloud users can also access storage outside the cloud, e.g. via NFS, xrootd [3], Gluster [4], and other protocols.

OpenStack services have been deployed in Padova, while compute nodes (where cloud virtual machines are instantiated) have been installed in both sites.

Concerning the hardware, in Padova a DELL Blade based solution was chosen. It includes:

- 4 E5-2609 servers with 8 cores and 32 GB of RAM, used to host the cloud services in high availability mode, as discussed in sec. 4;
- 5 E5-2670-v2 hosts, each one with 40 cores and 96 GB of RAM, used as cloud compute nodes;
- 5 E5-2650-v3 hosts, with 40 cores and 96 GB of RAM, used as cloud compute nodes.

The compute nodes in Legnaro are instead 6 E5-2650v2 Fujitsu Primergy RX300S8 servers, each one with 32 cores and 96 GB of memory.

Concerning the storage, in Padova a iSCSI DELL MD3620i server solution was chosen. It now includes 23x900GB SAS disks, but a storage expansion MD1200 with 12x4TB disks is being

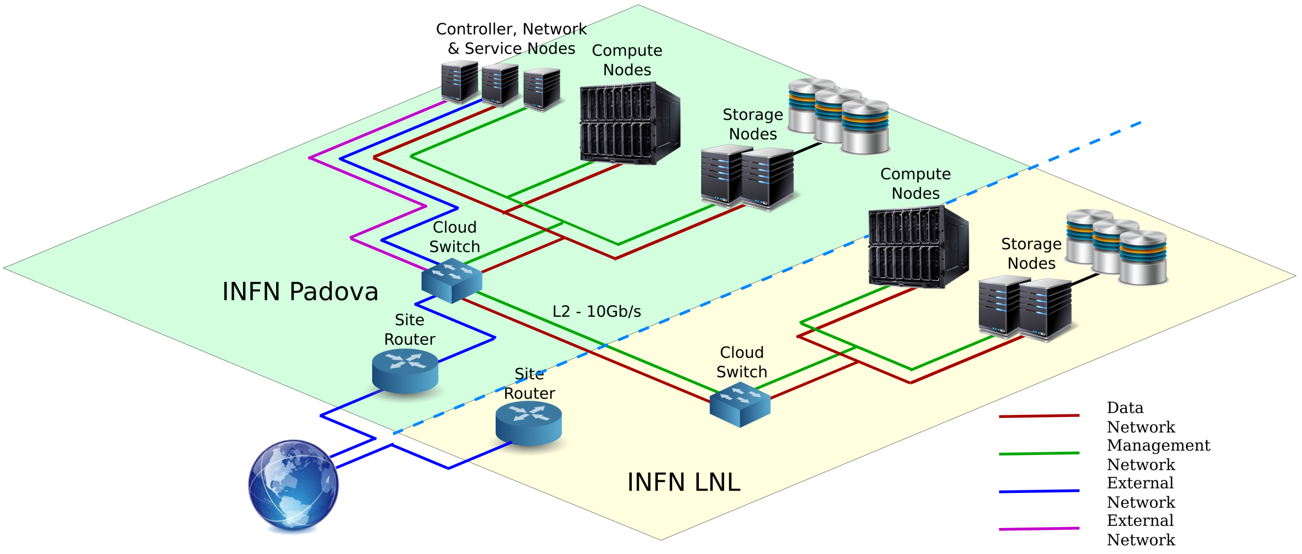

Figure 1. Layout of the Cloud Area Padovana

integrated. This storage system has been configured using GlusterFS, and is used for the cloud images (Glance service), for the storage of the virtual machines (Nova service) and for the block storage available to the cloud instances (Cinder service). In Legnaro a Fibre Channel based system (DELL PowerVault MD3600F plus MD1200 expansion) with 24x4TB disks was instead acquired. It will be used as general-purpose user storage, also configured using GlusterFS.

3. Network layout

The Padova and LNL sites are connected through two dedicated 10 Gbps Ethernet optical data links: one is used for the distributed Padova-Legnaro Tier-2 and the other one for all the other activities (so it includes also the Cloud Area Padovana workload).

The three "typical" cloud networks (management, data/tunnel and public/external networks) have been deployed. They have been implemented through three VLANs, each one associated to a class C network. These VLANs are shared between the two sites. OpenStack Neutron service with Open vSwitch and GRE (Generic Router Encapsulation) [5] is used.

OpenStack networking has been configured to allow the access to cloud Virtual Machines from the INFN-Padova and INFN-LNL LANs without the need of public (floating) IPs.

This has been implemented with two Neutron provider routers. The first one is used for virtual machines that need to be reachable from outside; it has the external gateway on the public network, and connects the 10.63.x.0 class C networks (each OpenStack project uses a separate network). In this case virtual machines can access the Internet through the NAT provided by Neutron, and can obtain a floating IP from the public network in case they need to expose services to the external world.

By contrast, the second router connects the 10.64.x.0 class C networks (one per OpenStack project), but has the external gateway towards the internal LANs. The virtual machines connected through this router cannot obtain a floating IP address, but are still allowed to access the Internet through an external NAT.

A special configuration is also used to integrate storage systems external to the cloud, to provide virtual machines with high performance access to data located in these storage servers. This was implemented deploying the L2 Neutron Agent software on the storage servers, so that network bridges and GRE tunnels towards all the components of the cloud infrastructure are automatically created when services start. The only manual operation needed is to add the new traffic flow to the local OpenFlow table. As this table is restored when new compute nodes are added to the cloud, a cron job is used to keep it updated.

4. Deployment of Cloud services

Cloud services have been deployed in high availability mode using four servers: two hosts have been configured as Controller nodes and two as Network nodes.

High availability for the OpenStack services has been implemented considering the active/active mode, relying on the HAProxy [6] and Keepalived [7] services, which also provide load balancing capabilities.

The HAProxy and Keepalived services are also used for the MySQL database, implemented through a Percona XtraDB cluster [8], composed of three instances. The database cluster has been configured in multi-master (active/active) mode: all Percona instances can receive queries, and keep their local databases synchronized with each other by means of the Galera libraries.

The HaProxy/Keepalived cluster is implemented by three virtual machines, hosted on three different hypervisors on a Proxmox local facility. This Proxmox cluster is now used also for the three MySQL Percona instances, but the migration of these hosts to physical nodes is on going.

Fig. 2 shows how the services have been deployed in High Availability mode.

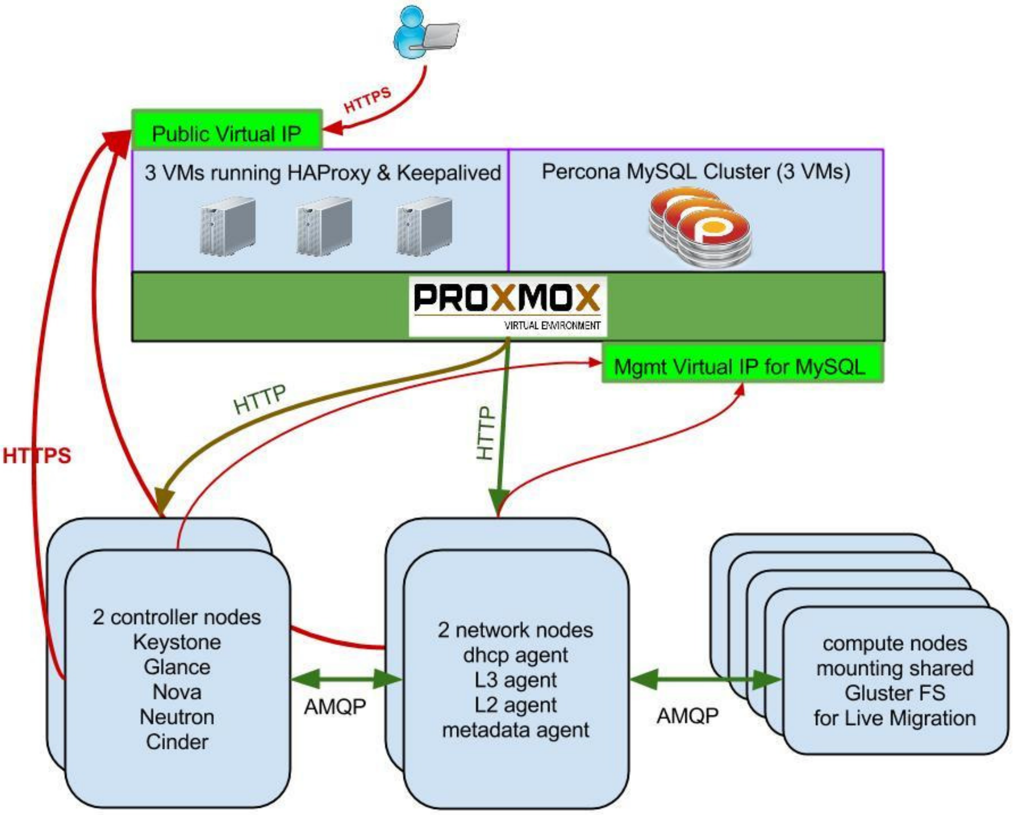

Figure 2. Deployment of services in High Availability mode in the Cloud Area Padovana

The OpenStack components running on the controller nodes are: Keystone, Glance, Nova-API, Nova-NoVNCProxy, Nova conductor, Nova consoleauth, Nova Cert, Nova scheduler, Neutron server, Cinder-API, Cinder-Volume and Cinder scheduler. OpenStack services communicate through the AMQP (implemented using RabbitMQ) daemon running in the controller nodes and configured in high availability mode.

By contrast, the network nodes run the Neutron agents (dhcp, L3, metadata, openvswitch). All of them, with the exception of L3, can run in active/active mode because they're stateless and communicate by means of the highly available AMQP daemon. If an L3 becomes unavailable, the virtual Neutron routers handled by it can be migrated to the other L3 instance running on the other network node.

Services are exposed through secured SSL endpoints, even if each component actually listens in plain HTTP. This configuration has been implemented through HAProxy, which has been configured to act as an SSL-terminator. The incoming connections (from clients and from OpenStack services) are encrypted and directed to HTTPS URIs: HAProxy is then responsible for extracting the plain payload and forwarding it to the relevant OpenStack API.

Concerning the storage, all the hypervisors mount a shared GlusterFS file system for the instances, and this allows the live migration of virtual machines between different compute nodes, if needed. GlusterFS is also used for the image (Glance) and block storage (Cinder) backends. In the current implementation the GlusterFS servers are hosted in the controller nodes. Separation of storage services with cloud services is being implemented since the current implementation proved to be not too reliable.

5. Authentication and authorization

An important development that has been done in the Cloud Area Padovana is the enhancement of the authentication and authorization module of OpenStack that provides basically only the classic username/password mechanism.

Extensions to the OpenStack identity service (Keystone) and web portal (Horizon) were implemented to integrate with SAML or OpenId based identity services.

In particular the integration with the SAML compliant *INFN Authentication and Authorization Infrastructure* (INFN-AAI) was implemented. This means that INFN users can authenticate to the Cloud Area Padovana using the same procedure used to access other INFN services.

In addition, a new extra component to manage the registration workflow has been implemented. The registration workflow involves different subjects, each one with different roles and capabilities. In the current implementation two categories of actors are defined: the cloud administrator and the project administrator. The cloud administrator has all the privileges of users and projects and is responsible, eventually, for accepting or rejecting the request for registration whereas the project administrator is the project user who is responsible for validating any new request for registration for that project.

The complete registration workflow is depicted in Fig. 3.

This registration flow was implemented in the OpenStack Horizon Dashboard through the following elements:

- A user registration form to allow users to specify the projects to subscribe to or to be created.
- A panel for the registration request management, visible only by the cloud administrator, and a set of pop-up menus dealing with different steps of the flow.
- A panel by which the project administrator can handle the subscription requests for the relevant project.

These elements are registered into the web portal exploiting the extension mechanisms of the framework. No changes in the existing code of the Horizon Dashboard were needed.

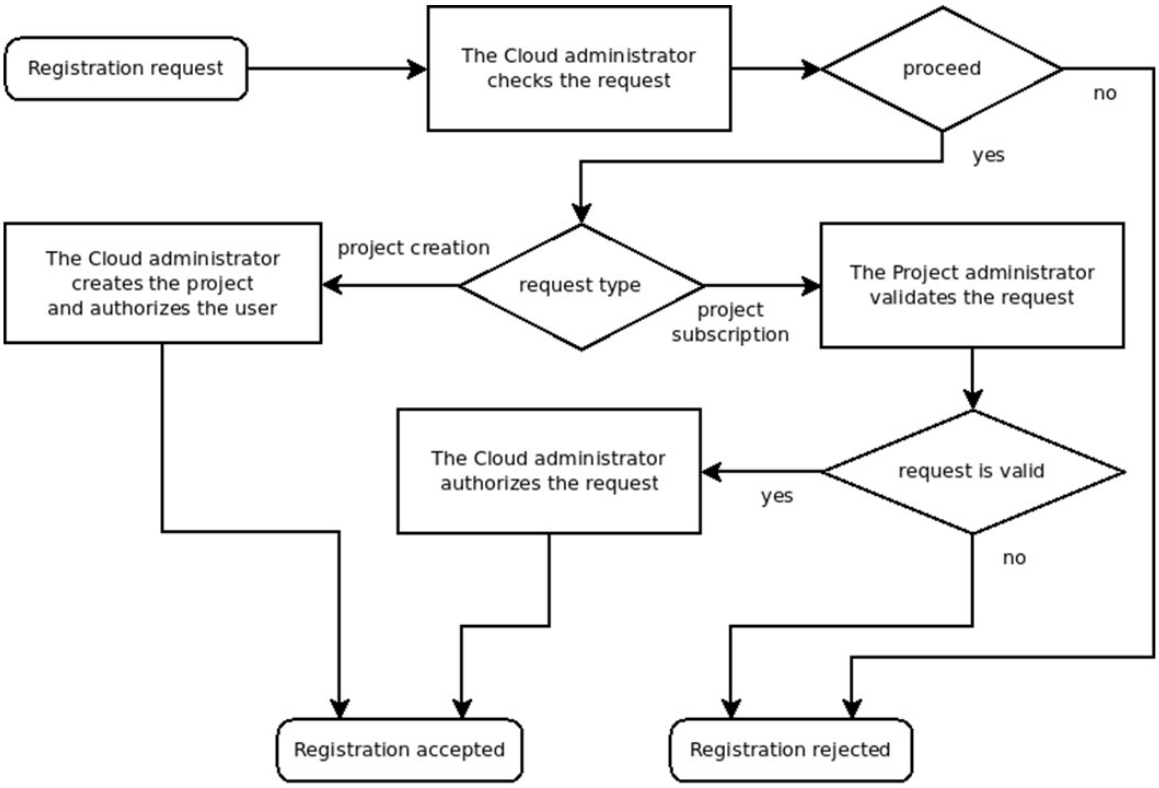
Registration request
The Cloud administrator checks the request
proceed
no
yes
The Cloud administrator creates the project and authorizes the user
project creation
request type
project subscription
The Project administrator validates the request
The Cloud administrator authorizes the request
yes
request is valid
no
Registration accepted
Registration rejected

Figure 3. Registration workflow in the Cloud Area Padovana

6. The fair-share scheduler

In OpenStack based clouds, the resource allocation to the user teams (i.e. the projects) can be done only by setting fixed quotas. Such an amount of resources cannot be exceeded by one group even if there are unused resources allocated to other groups. So, in a scenario of full resource usage for a specific project, new requests are simply rejected. This static partitioning of resources usually means a low efficiency in the overall resource usage.

To address this issue, work is on-going to enhance the OpenStack scheduler capabilities by providing:

- a fair-share based resource provisioning model (based on the SLURM Multifactor Priority Strategy [9]) to guarantee that the resource usage is equally distributed among users and groups by considering the portion of the resources allocated to them (i.e. share) and the resources already consumed;
- a persistent queuing mechanism for handling also the requests that can't be immediately fulfilled.

The chosen approach was to implement a new component named *FairShare-Scheduler* which aims to provide the missing advanced scheduling logic without breaking the OpenStack architecture. The schema in Fig. 4 shows the high level architecture and in particular highlights the interaction with other components involved in the scheduling activity.

A proper priority value, calculated by the FairShare-Manager, is assigned to every user request. The request is immediately inserted in a persistent priority queue by the PriorityQueue-Manager. A pool of Workers then fetches from the queue the requests having highest priority and sends them in parallel to the Nova scheduler through the AMQP messaging system. The

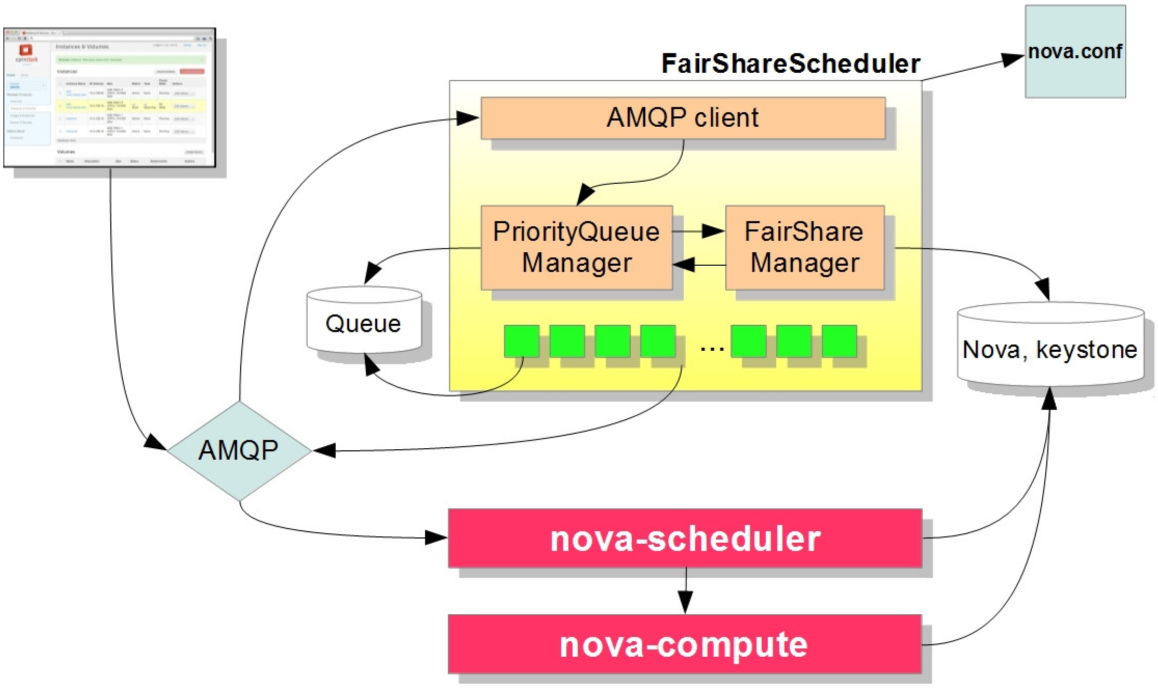

Figure 4. Architecture of FairShare scheduling

priority of the queued requests is periodically recalculated to give a chance to the older requests to rise up in the queue.

In the scenario of full resource utilization, the pool waits until some compute resources become available.

To prevent any possible interaction issue with the OpenStack clients, no new states have been added: from the client point of view the queued requests stay in “scheduling” state till the compute resources are available.

A prototype of this fair-share scheduler has been implemented and it is now being validated.

7. Operations

Foreman [10] and Puppet [11] are the installation and configuration tools that have been chosen for the Cloud Area Padovana. Foreman is a system that allows the automatic installation of the operating system, and then coordinates the configuration of all services.

Configuration of several services have been automated using Puppet modules. In particular the configuration of all the services on the cloud compute nodes is completely implemented through Puppet. Some of these Puppet modules have been found on public available repositories (e.g. Puppet forge), while others are in-house implementations.

The monitoring of the Cloud Area Padovana is done using the Ganglia [12] and Nagios [13] tools. While Ganglia is mainly used to monitor the status and performance of the resources, Nagios is used to check the availability, functionality and performance of the overall infrastructure, and to notify the cloud operators in case of problems. It is used not only to monitor the physical resources but also to control all the services, such as the OpenStack components, the database, the storage systems, etc. Specific custom Nagios sensors were developed already if not available. Basically every time that a new problem in the operations is found, a specific new Nagios sensor is implemented to prevent the problem in the future or at least to detect it early.

8. Usage of the Cloud Area Padovana

At the time of writing more than 60 users belonging to about 15 projects are registered in the Cloud Area Padovana.

ALICE [14] and CMS [15] are among the experiments that are using/testing this cloud infrastructure.

For the ALICE team the main use-case was to implement on the cloud a Virtual Analysis Facility (VAF) based on virtual machines. The idea is the creation of an elastic cluster for the interactive analysis: resources are allocated and released in dynamic way. The user builds his cluster asking for resources with a simple shell command, releasing them when the work is done. In this VAF the workload management system is implemented using HTCondor.

However, for the CMS team the cloud is being evaluated as a replacement of a local facility, mainly used for interactive processes, which is currently implemented by a limited set of physical machines. In these interactive processings, input data is usually read from the dCache grid storage located in Legnaro, while results are written to a Lustre based storage system located in Padova. This Lustre storage also hosts the user's home directories. These machines also work as LSF clients (to submit batch jobs to the Tier-2 LSF cluster) and act as grid User Interfaces too. To reproduce this complex environment on the cloud, a specific image was provided. The network was also configured so that the CMS Virtual Machines on the cloud can access the Tier-2 network.

9. Conclusions

We described in this paper the Cloud Area Padovana, a IaaS providing computing and storage resources on demand to several experiments and other research teams.

The cloud is configured in high availability and load balanced mode.

Some in-house developments have been integrated, such as the one to enable the access to the cloud through SAML compliant federated identity providers, and the module to manage user registrations.

A fair-share scheduling mechanism is under development, to overcome the static partitioning of resources that would lead to a poor utilization of resources.

The cloud is expected to expand its overall capacity offer by including new hardware resources that will be acquired by the research teams to address their increasing computing needs.

References

[1] M. Biasotto et al., *The Legnaro-Padova distributed Tier-2: challenges and results*, J.Phys.Conf.Ser. 513 (2014) 032090
[2] Home page for the OpenStack project, http://www.openstack.org
[3] A. Dorigo et al., *XROOTD- A highly scalable architecture for data access* WSEAS Trans. Comput. 1(4.3) 2005
[4] The Gluster web site, http://www.gluster.org/
[5] S. Hanks et al., *Generic routing encapsulation (GRE)* (2000). http://tools.ietf.org/html/rfc2784.html
[6] W. Tarreau, *HAProxy-The Reliable, High-Performance TCP/HTTP Load Balancer*, http://haproxy.1wt.eu
[7] A. Cassen, *Keepalived: Health checking for LVS and high availability*, (2002), http://www.linuxvirtualserver.org
[8] Percona XtraDB Cluster, https://www.percona.com/software/mysql-database/percona-xtradb-cluster
[9] Home page for the Multifactor Priority Plugin, https://computing.llnl.gov/linux/slurm/priority_multifactor.html
[10] Foreman home page, http://theforeman.org/
[11] Puppet Labs home page, https://puppetlabs.com/
[12] M. Massie, B. Chun, D. Culler, *The Ganglia Distributed Monitoring System: Design, Implementation, and Experience*, Journal of Parallel Computing, vol. 30, no. 7, July 2004.
[13] W. Barth, *Nagios. System and Network Monitoring, No Starch Press, u.s (2006)*
[14] K. Aamodt et al., The ALICE experiment at the CERN LHC, Journal of Instrumentation 3.08 (2008): S08002.
[15] S Chatrchyan et al., The CMS experiment at the CERN LHC, Journal of Instrumentation 3.08 (2008): S08004.