Use cases, architecture and roadmap for the INFN Corporate Cloud

Stefano Stalio, Cristina Aiftimiei, Marica Antonacci, Antonio Budano, Giacinto Donvito, Claudio Grandi, Matteo Panella, Davide Salomoni, Vincenzo Spinoso, Riccardo Veraldi, Federico Zani, Stefano Zani

Overview
The INFN Corporate Cloud
Features and user experience
Use cases for the INFN Corporate Cloud
    Local and central computing Services
        Example of a distributed web application
        DropBox-like sync and share
    Scientific Computing
Architecture
    Network architecture
        OpenStack “Management Network”, L3 and cross-site
        The cloud.infn.it DNS domain
    Identity Service
    Distributed Object Storage
        Deployment of the Swift Object Storage in the INFN Corporate Cloud
        Features
    Image Service
        Features
        Cross-cloud image management
    Block Storage
        Distributed Block storage
        Features
    Monitoring
    FWaaS, LBaaS, VPNaaS and Resource Orchestration
    Syslog
        Tracing user activities
    Infrastructure automation
        Puppet / Foreman
        Docker
Management
    Requirements for member sites
    Management model
Roadmap

Overview

The INFN Cloud Working Group has now been active for almost three years within the INFN Commissione Calcolo e Reti (CCR), its activity being that of testing and acquiring expertise on technologies related to Cloud Computing and of selecting solutions that can be adopted in INFN sites in order to meet the computing needs of the INFN scientific community and more generally to ease information sharing inside (and maybe outside) INFN.
A number of projects related to Cloud Computing started in INFN thanks to the knowledge and expertise that were the outcome of the activity of the Cloud Working Group.
A restricted team has been working in the last two years on the deployment of a distributed private cloud infrastructure to be hosted in a limited number of INFN sites. The INFN Corporate Cloud (INFN-CC) working group has been planning and testing possible architectural designs for the implementation of a distributed private cloud infrastructure and implemented a prototype that is described in this document.

The INFN Corporate Cloud

INFN-CC is intended to represent a part of the INFN Cloud infrastructure, with peculiar characteristics that make it the optimal cloud facility for a number of use cases that are of great importance for INFN.
While the INFN Cloud ecosystem will be able to federate heterogeneous installations that will forcibly adopt a loose coupling scheme, INFN-CC tightly couples a few homogeneous OpenStack installations that share a number of services, while being independent, but still coordinated, on other aspects.
The focus of INFN-CC is on **resource replication, distribution and high availability**, both for network services and for user applications.
INFN-CC represents a single, though distributed, administrative domain.

Features and user experience

The main, peculiar features of INFN-CC are:

- single point of access to distributed resources, fully exploiting the native functionalities of OpenStack and with no need of external integration tools.

- SSO and common authorization platform. User roles and projects are the same throughout the infrastructure, while quotas for projects vary from site to site.

- common DNS name space for distributed resources. DNS HA provides high availability for distributed resources.

- secure dashboard and API access to all services for all users. The dashboard, os apis and ec2 apis are available.
  All services are implemented on top of an SSL layer, in order to secure resource access and data privacy.

- easy sharing of VM images and snapshots through a common Object Storage deployment.
  A single image/snapshot database is used by all the project sites. This means that all VM images and snapshots are available in all sites.

- block device sharing over remote sites;
  A rough way to implement is through CEPH or SWIFT backend volume backups, faster and more efficient ways are under investigation.

- self-service backup for instances and block storage. Backed-up data can be accessed/restored transparently from/to any site.
  Final users and tenant administrators are responsible for backing up their instances and the attached block devices. Adequate tools, native to OpenStack, are provided. As the backup storage backend, both for instance images and snapshots and for block devices, are replicated and distributed, data backup is transparently available in all the cloud sites and is still available in the case of a site failure.

- INFN-CC is seen as a single infrastructure by federated Clouds, both belonging to INFN and to other institutions.

- PaaS could be deployed in top of the INFN-CC, but today the focus is on IaaS.
  The INFN-CC aims at providing an IaaS infrastructure for INFN. One or more PaaS platforms might, in the future, exploit the existing IaaS platform to improve and ease the user experience.

Use cases for the INFN Corporate Cloud

The architecture of INFN-CC is particularly fit for a wide range of use cases where a strict relation exists between resources that are distributed over different sites.
Most of these use cases are related to the delivery of computing services for the INFN community, be they of local interest for users belonging to a single INFN site or of general interest for the whole community.
This does not mean that scientific computing is unfit for the INFN-CC, but often scientific computing environments do not need the high availability features provided by INFN-CC and can take advantage of other cloud deployments.

Local and central computing Services

The distributed architecture of INFN-CC is the natural environment for the easy deployment of distributed network applications with high availability standards, **where the failure one or even two whole sites is unnoticed by users**.

Such deployments can benefit from the adoption of database clustering tools (e.g. Percona Xtradb cluster), load balancers (HAProxy), distributed storage systems (Swift, CEPH), monitoring systems (Nagios, Zabbix). These can easily be used or deployed, provided security and legal constraints are respected, on a distributed cloud environment by the very end users, like system administrators belonging to any INFN site or working for an experiment or project.
INFN-CC also relies on the features provided by the INFN DNS HA national service for granting uninterrupted availability of its internal services.
Some of the OpenStack features, like the orchestration tools, VPNaaS, LBaaS, FWaaS, make the work of people who want to use the cloud infrastructure for their applications even easier.

Given these premises, INFN-CC is the best tool to use for the distributed, highly available deployment of:

- web sites and portals;
- web based applications;
- information/documentation/data sharing tools;
- authentication/authorization services (kerberos, ldap, radius servers, IdPs);
- mail services, where performance requirements are satisfied on a virtualized environment;
- database services;
- room booking, calendars;
- test and development environments;
- educational and training environments.

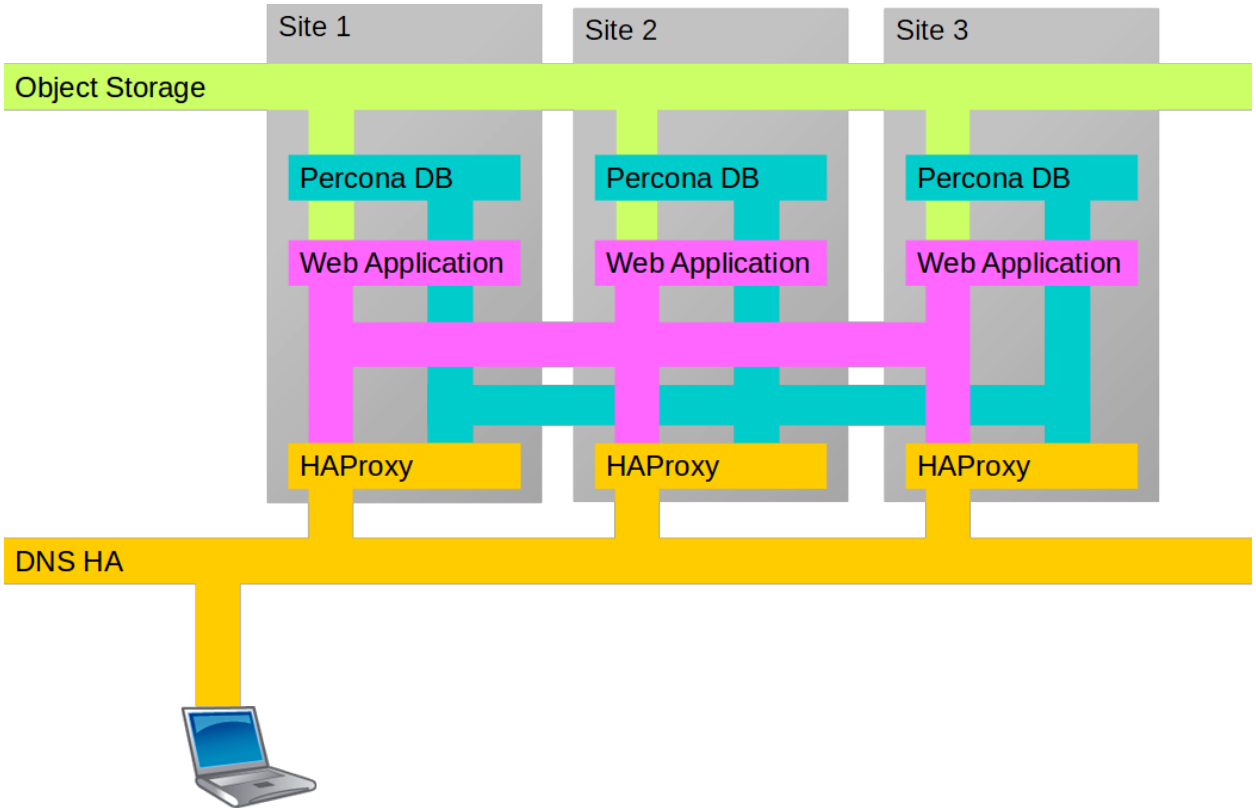
The “Cloud approach” represents a paradigm shift in the way network services are implemented and delivered and a giant step forward towards service and data availability.

Also, the fact that people will be working on a distributed infrastructure will blur the today sharp borders between “local computing services” and “central computing services”, because the implementation model is the same, the audience being the only remaining difference.

Example of a distributed web application

The figure below shows an example of a generic distributed web application implemented on INFN-CC. This application uses a SQL database, accessed through HAProxy, and an object storage data backend. There is no single point of failure and each component is redundant

and distributed. The failure of one or even two sites does not affect final users, that are still able to use the application without noticing anything but a possible performance degradation.
While a distributed object storage backend for application data is deployed rather easily and already available on INFN-CC, posix access to data from multiple, distributed hosts is not easily and reliably obtained without a low latency link and special file system setups.
The example below might not be economically convenient, in term of resources needed (e.g. number of VM instances), for the provisioning of a single application, but the same setup can be used by more than one application and the availability level can be reduced by lowering to two the number of load balancers and application servers, while the DB cluster needs an odd number of hosts to work correctly.



DropBox-like sync and share

An INFN-wide DropBox-like sync and share infrastructure relying both on a web interface and on a sync client application would be a powerful tool for improving communication among INFN scientists and staff, overcoming quota limitations that represent the main limit of commercial platforms and granting INFN full control over shared data.
Such an infrastructure might easily be hosted by INFN-CC.
The application servers and database servers would be redundant and load-balanced VM instances distributed across the cloud sites, while a centrally managed, but resource-wise distributed, Swift Object Storage might act as the storage back-end.

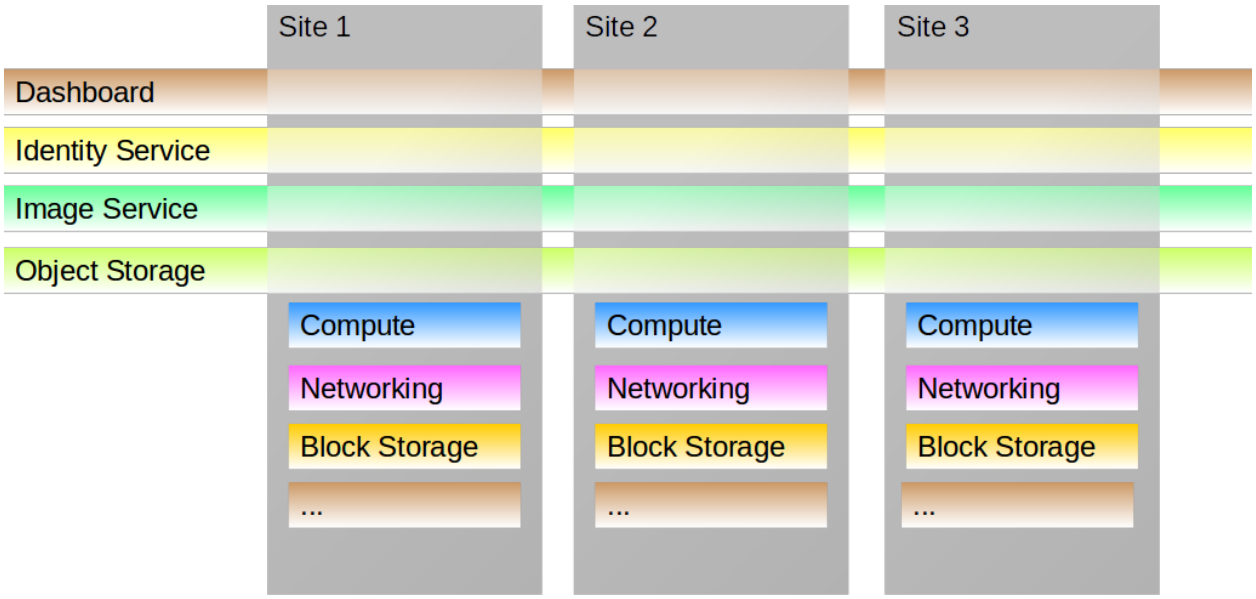
Two example applications that might provide this service are Pydio (https://pyd.io/) and OwnCloud (www.owncloud.org), that are in use in many INFN sites, are known to be able to take advantage of SQL clustering tools and load balancers for high availability and can use a Swift back-end for user data.
Both Pydio and OwnCloud feature a multi-platform synchronization client and a user-friendly web interface.

Scientific Computing

Massive data analysis or simulations do not usually need an environment like that of INFN-CC, but this does not mean that the INFN-CC doors are closed to scientific computing.
Tier 3 virtualization, the last mile of data analysis, as well as software development environments are the first use cases that might take advantage of INFN-CC and use it efficiently.
Further use cases might be applicable in the future, according to the available resources and to the project development.

Architecture

INFN-CC is composed of different OpenStack installations sharing a set of services that are managed globally while maintaining other services local, as seen in the picture below.

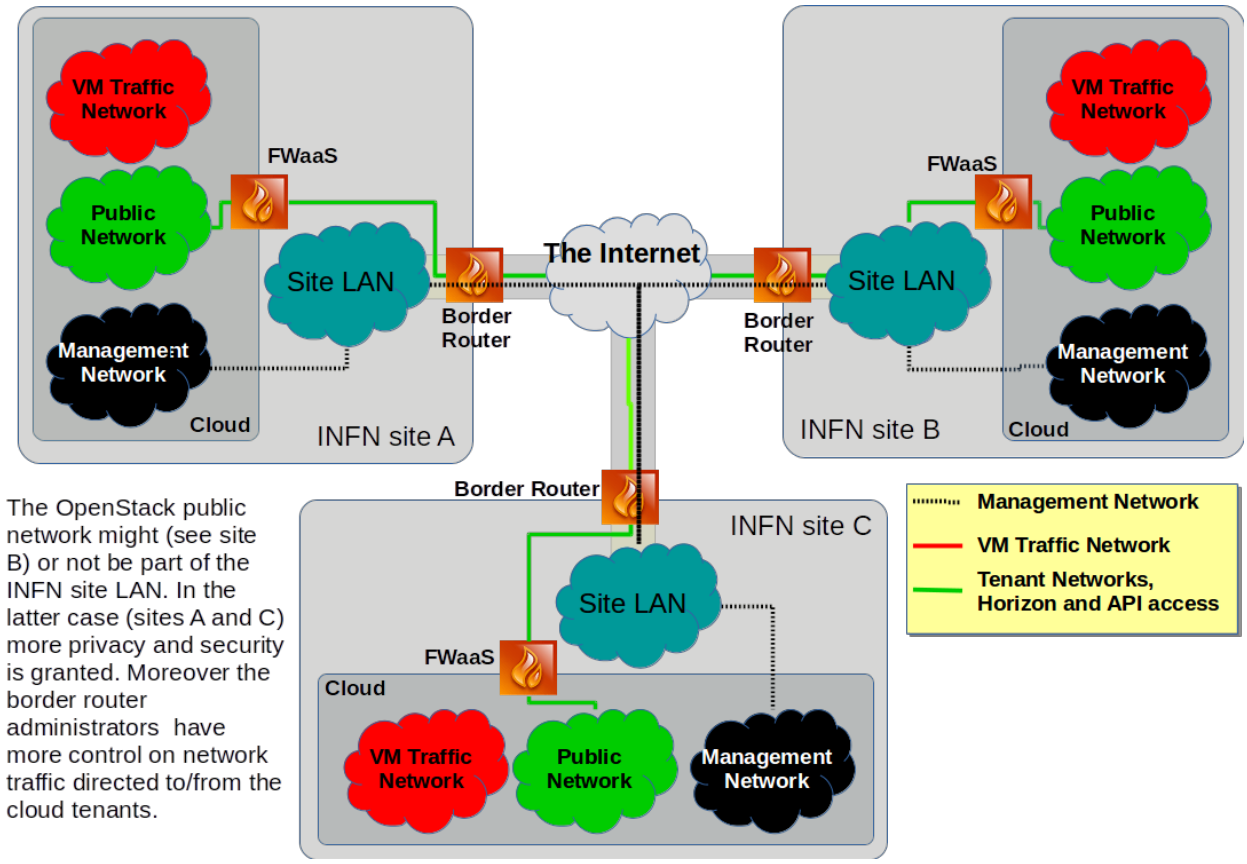


There must of course be an agreement on how the centralized services are deployed and managed and there must be a set of common features that the locally managed services must support, in order to give the final users an even environment to rely on.

Network architecture

OpenStack “Management Network”, L3 and cross-site

According to the OpenStack documentation, for a standard “per tenant router” network setup three different networks are required: a private network for VM internal traffic, another private network for management and a public network for user access to cloud resources.
In the INFN-CC model the VM networks remain private and do not cross the border of their own “region”. Also public networks are separate and managed locally, except they might benefit of a cross-site DNS domain namespace in order to allow for easy service migration.
Hosts on the management networks of the different sites, on the other hand, must be able to intercommunicate, possibly taking advantage of a set of loose firewall rules, in order to speed up the system setup and maintenance. Also, a cross-site DNS domain namespace would be helpful in order to dynamically migrate cloud services, when needed, for high availability.



The OpenStack public network might (see site B) or not be part of the INFN site LAN. In the latter case (sites A and C) more privacy and security is granted. Moreover the border router administrators have more control on network traffic directed to/from the cloud tenants.

The cloud.infn.it DNS domain

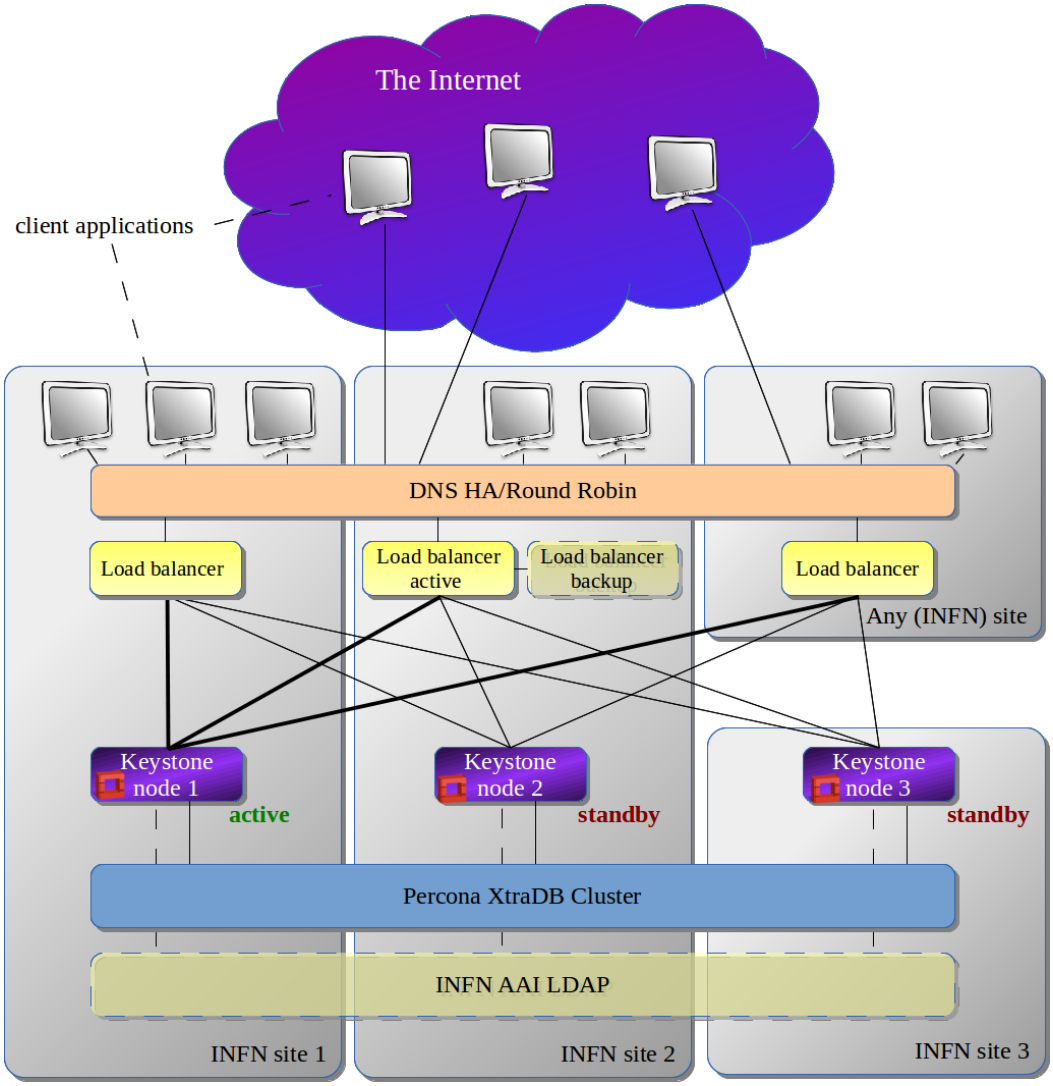
In a geographically distributed environment it is important to have the possibility to manage a “cross-region” domain, so to let applications and users locate services no matter where they are deployed.
Ideally the “cloud.infn.it” DNS records shall be managed both by humans and by applications (via api/scripts), in order to increase the overall level of infrastructure automation.
This will be achieved also by using the INFN DNS HA services, with a tight collaboration between the INFN-CC and the INFN DNS HA team.

INFN DNS HA is a powerful tool that, together with other instruments, will allow high availability both of services building the Cloud infrastructure and of services hosted by the Cloud infrastructure itself.

The first reason for having a “cloud.infn.it” DNS domain, unbound from a particular INFN site, is that of easing the implementation of high availability for the services needed by INFN-CC.
Nonetheless, also the model of application delivery described earlier in this document needs to rely on a DNS domain - be that “cloud.infn.it”, “infn.it” or anything else - that is not bound to a single site; in fact network applications implemented over INFN-CC can be delivered by different sites in different moments and there must be a way for clients to always direct their requests to the active servers.

Identity Service

Authentication and authorization services must guarantee uninterrupted availability, even if a whole site fails, and must thus be distributed and replicated on a geographic basis. OpenStack provides an integrated identity service, called Keystone, that relies on a SQL database, both for internal bookkeeping and as the default back-end for users, groups and roles.
LDAP can substitute SQL as an authentication and/or authorization back-end, but not for the storage of internal information.
A must for INFN-CC is the possibility, for any INFN associate, to use her/his AAI credentials to access cloud resources. For this reason the INFN AAI LDAP database is used as the authentication backend. At the same time full freedom in defining access policies (i.e user roles within projects) is granted by relying on the native Keystone MySQL backend for authorization.
In order to obtain a highly resilient authentication and authorization infrastructure, three Keystone servers have been instantiated in as many INFN-CC sites.
The SQL database is replicated on the three sites using the Percona XtraDB Cluster, a high availability and high scalability solution for MySQL clustering.
In order to increase performance - because the DB replica is synchronous - and to ensure data consistency, only one of the Keystone servers must be active at any moment, while a second one must be ready to take the place of the first should any problem occur. The third server is important in order to establish the cluster quorum and avoid split-brain situations, besides being ready to become the active server in the very unlikely case the first two will be unavailable.



HAProxy is used as a load balancer and HA provider. One or more HAProxy nodes contact the active Keystone server, falling back to the second and to the third should the checks they periodically perform on the active authentication server fail.
High availability of the proxy nodes on a geographic basis is obtained using a multi-master, dynamic DNS (INFN DNS HA) that can automatically update its maps according to the sanity checks it performs on the nodes for which it provides name to address translation.

Distributed Object Storage

Similarly to what happens for authentication and authorization, there is a need for a storage infrastructure capable of transparently replicating data over geographically distributed sites and providing reasonably fast access from distributed endpoints.
Recent tests proved that POSIX file systems (glusterFS, Lustre, . . .) are unfit for the realization of a geographically distributed, redundant and highly resilient storage facility, while

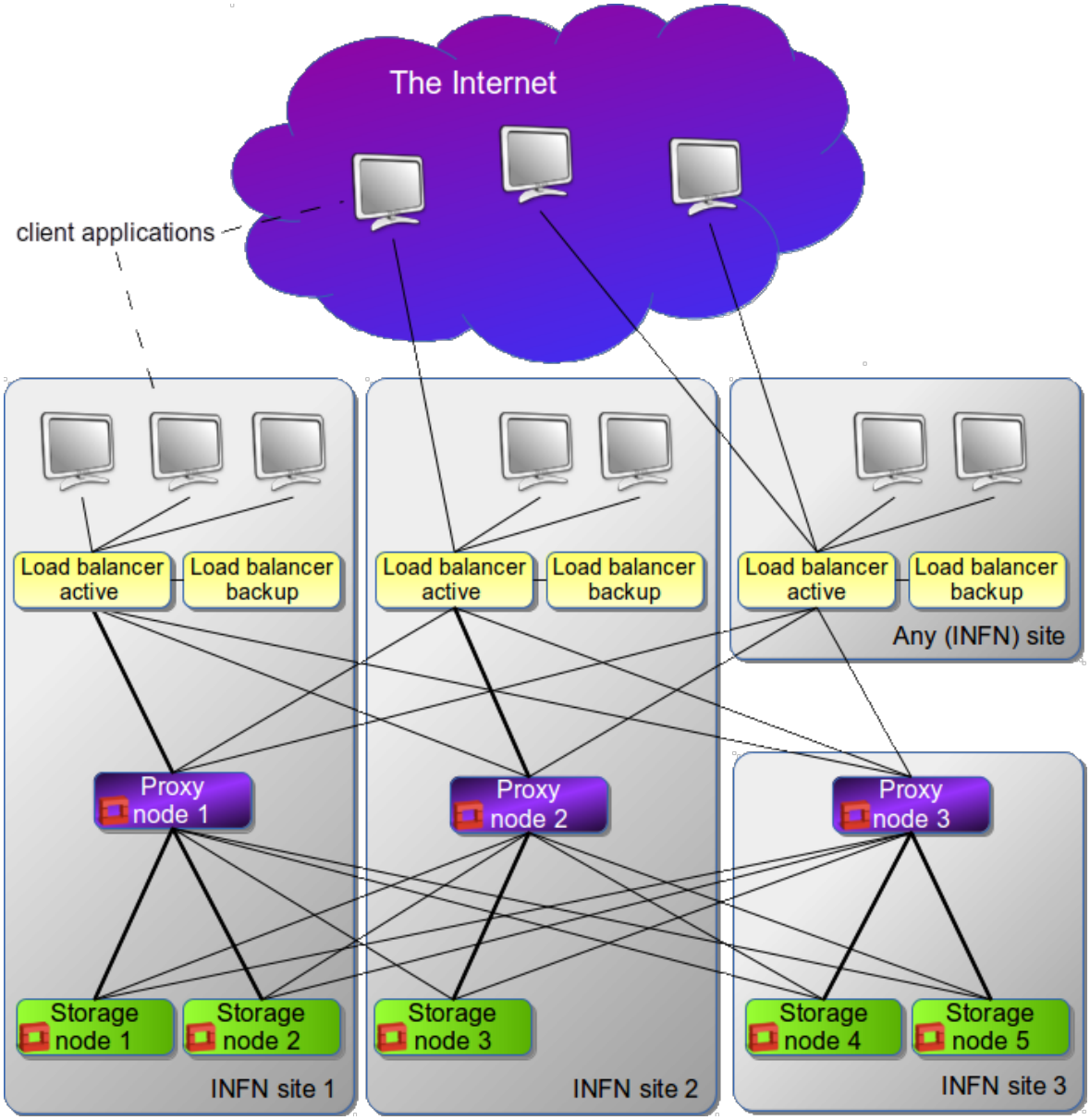
object storage systems that use higher level, TCP based, transmission protocols for internal communication and for the user interface seem to be best fit for this task.
The OpenStack Swift object storage has all these characteristics and can be successfully deployed in a geographically distributed environment. A working setup exists today in the INFN-CC sites.
A distributed object storage infrastructure based on Swift, and thus fully integrated in OpenStack, is available for:

- VM image archiving/distribution/sharing across sites;
- Block device backup and distribution across sites;
- personal data;
- application data (e.g. images and videos for web sites);
- scientific data;
- sync&share storage backend.

Also, a distributed Object Storage infrastructure might help INFN experimental collaborations in a critical task that has often been solved with non-standard, home-made approaches: that of distributing scientific data in other INFN sites or in other countries in order to guarantee data redundancy and availability.

Deployment of the Swift Object Storage in the INFN Corporate Cloud

Each INFN-CC site represents a Swift “region”, while inside each region, Swift storage servers may be logically divided in availability “zones”. This hierarchic structure is used by the Swift replica engine in order to optimize replica location and thus data availability. Swift proxy services are employed to provide data access to the end user, while the authentication and authorization services are provided by the OpenStack Keystone authentication provider.
Also the proxy servers are replicated geographically, allowing users to access data from a “near” site and contributing to the availability and resilience of the system. HAProxy is used to provide load balancing and HA for the proxy servers (Figure 3) .



Features

User must be able to

- create/delete containers and objects;
- use the OpenStack Dashboard interface to the OSS
- use native APIs
- write/use customized programmatic interfaces to the OSS;
- select which storage policies, among those available, they want to apply to their containers
- possibly define storage policies

Image Service

Virtual machine images sharing among sites is very important: having a central repository ensures that a coherent set of images is always available on all the different sites. Images may be revised, for example to integrate new security patches and immediately made available on the whole infrastructure.
The OpenStack Image Service, provided by Glance, is implemented on top of a Swift Object Storage Backend. This eases image management, guarantees data availability and allows for transparent cross-site distribution of VM images and snapshots.
As the Image Service is one and managed centrally, al VM images and snapshots are always and immediately accessible from any INFN-CC site.

Features

Users are be able to:

- create/start instances;
- reboot instances;
- stop instances;
- terminate instances;
- snapshot instances;
- backup instances;
  same as snapshot but can be programmed and the number of versions to keep can be controlled.
- start the same public or private images and snapshots in any cloud site.

Administrator must also be able to

- cold migrate instances;
- possibly hot migrate instances.

Cross-cloud image management

As for cross-cloud image management, the VMcaster/VMcatcher is an interesting tool-set that is currently the recommended image management system in the EGI Federated Cloud.
It uses an internal SQLite database where image information and lists are stored.
All image lists are signed by an authenticated endorser with his/her personal X.509 certificate. This means all images are referenced by a Virtual Machine Image List which contains a secure hash (SHA512) signed using X.509 personal certificates (provided by the image list endorser). These Virtual Machine Image Lists are published, and interested sites subscribe to the Lists in the resulting catalogue.
Another important feature of VMcaster/VMcatcher is the support of Cloud framework agnostic tools (plugins are available for the most used CMF, like Openstack and OpenNebula).

Another interesting tool is Glint, an image replication system developed by the High Energy Physics Lab at the University of Victoria in British Columbia Canada. It is based on Glance,

the Openstack Image Service, and allows to distribute and synchronize images among different Glance repositories.

Block Storage

The block storage service, provided by Cinder, is deployed locally using Ceph (http://ceph.com). Because block storage backup exploits the distributed Object Storage infrastructure described in this document, volume backups are available in all sites to be restored and reused.

Distributed Block storage

This is still an open issue.

Features

Users must be able to:

- create volumes;
- attach volumes to their instances;
- detach volumes from their instances;
- backup volumes;
- access and restore backups to a different site;
- share volumes (to be investigated);
- access their volumes from a different site (to be investigated).

Monitoring

Because an OpenStack setup is a complex ecosystem, and a multiregional OpenStack is even more complex and rich of interdependent components, monitoring it efficiently is at the same time extremely important and challenging.

Checks must be performed at multiple levels:

- hardware and infrastructure
  UPS and power distribution, temperature, network infrastructure, DNS and basic network services, storage infrastructure, power supplies...
- os and hypervisor
  on hypervisors and hosts providing cloud services: network connectivity, CPU load, memory usage, available disk space, ….
- OpenStack and related services

The lower level infrastructure monitoring is usually already present in an existing data center, and monitoring the OS on hypervisors and other servers is only a matter of expanding existing sets of checks to new hardware.

On the other hand, a whole new set of high level checks must be applied or implemented in order to monitor how OpenStack services and auxiliary services (e.g. Percona XtraDB Cluster, the ceph storage infrastructure, the LDAP backend) are behaving.
These checks can be as simple as verifying that a certain process is active or as sophisticated as creating a block device or an instance, making sure they are usable and deleting them. Not to be forgotten: SSL certificate validity is to be checked periodically by the same system that monitors the whole infrastructure in order to let administrators always ask for renewal in time.

FWaaS, LBaaS, VPNaaS and Resource Orchestration

FwaaS, LBaaS and VPNaaS are implemented locally in each site, although VPNaaS can be used to interconnect resources (i.e. tenant routers) that are hosted in different INFN-CC sites.
Resource Orchestration (Heat) has not been implemented yet, but will be in the future. Presently Heat is not capable of deploying resources on multiple regions. This limitation reduces its usefulness for the INFN-CC.

Syslog

In order to help cloud and tenant administrators to easily debug infrastructure related issues, a centralized log service has been deployed that receives logs from each middleware component: this may also include services not directly related to OpenStack, but participating in the infrastructure.
Since OpenStack uses rsyslog (which transmits all in clear-text over the wire) for managing log files, the use of an SSL layer is mandatory between central log server and producers.

Log files are accessible through a web interface, organized by host and date: the possibility to search log entries using a search engine (i.e. Solr or Elasticsearch) needs to be evaluated further. A possible solution would be employing rsyslog for log collection, Elasticsearch as a search engine and Kibana as a web GUI.
While for storing recent entries it might be useful to use log server's local/net attached disks, middle and long term log storage could exploit the built-in Object Storage system.

Depending of the log data size, an alternative approach would be that of implementing a hierarchical infrastructure where each site has a central log server collecting information from local resources and where a second level logging server, possibly replicated, only collects a subset of the logs from all sites.

Tracing user activities

Logging the system status with the purpose of debugging infrastructure and middleware issues does not close the syslog issue.
A much more challenging problem is that of logging user activities to obtain compliance with italian laws and with INFN and GARR rules for accessing network and computing resources.

INFN and GARR want to be able to trace who is responsible for each network connection exploiting their infrastructures and have been relying, for this purpose, to system logs provided by routers, DHCP and NAT providers, authentication services and so on.
OpenStack does not offer an easily readable logging system for tracking user activities, its logs being mainly meant for infrastructure debugging.
Extracting the information we need from there is not an easy task and further work is required to reach this goal.
On the other hand the new computing model brought up by cloud technologies must be accompanied by a new responsibility scheme. Most probably tenant administrators should be responsible for network activities on the resources they manage, no matter where they are located, rather than the IT staff maintaining the infrastructure.

Infrastructure automation

Puppet / Foreman

Every server (bare metal, virtual or dockerized) participating in the OpenStack infrastructure will be connected to a central Puppet master, which will deliver the right configuration.

The typical workflow would be :

- Cloud admins write and test puppet code on their local workstation
- Updates are pushed to INFN git repo (https://baltig.infn.it/fzani/cloud)
- Updates are automatically deployed to puppet master
- Associations between puppet classes, nodes, nodes roles and hosts groups will be managed directly though Foreman web interface

This will ensure versioning of the infrastructure configuration and the possibility to rollback or reproduce a specific system state.

Docker

Whenever possible - and whenever it makes sense - the docker ecosystem will be used to host services, to strongly decouple them and to simplify their management and replication.

A private and protected docker registry will be deployed to maintain, version and share docker images: possibly its backend will take advantage of swift or any other distributed file system provided by the cloud itself.

Management of docker ecosystem (hosts, images, containers and repo) might be easily automated using Foreman web interface.

Management

Requirements for member sites

INFN-CC requires homogeneous setups in order to ease administration and in order to improve the user experience. All sites belonging to INFN-CC guarantee that their OpenStack setup have the following features and provide the following services:

- Ubuntu as OS for the cloud infrastructure (to be confirmed)

- Common dashboard
- Common Identity Service
- Common Object Storage Services
- Common Image Service
- Common, distributed object storage back-end for images and snapshots
- Common, distributed object storage back-end for user data
- Common, distributed object storage back-end for block storage backup

- Block storage service
- “Per tenant network” model. This model becomes “per tenant and per region network” in a multi-region cloud environment.
- FWaaS
- LBaaS
- VPNaaS
- Orchestration Service

All member sites need to cooperate in administering the centralized Openstack Services: Identity, Object Storage, Image.

Management model

As resources in INFN-CC are so closely coupled and interdependent, they must be managed carefully by expert staff and must always work correctly.
For this reasons a limited number of cloud administrator, no matter where they are based, are allowed to administer hosts offering OpenStack services in any INFN-CC site both for normal maintenance and in case of emergency.
This approach is eased by the homogeneity of the infrastructure, but requires a trust agreement that breaks the barriers of the single site: **remote administrators must be trusted exactly as residents**.

On the other hand, infrastructure and hardware management is not easily performed from a remote site and should be full responsibility of the local IT staff. Cloud administrators and local IT staff should of course interact for better problem detection and solving.

This management model brings issues that exceed the technical and organizational problems of a distributed management team: hosting sites must agree on having external people manage part of their infrastructure as if they were local staff.

Roadmap

The goal of the INFN-CC working group is to have a working and functional prototype where users will be able to test their use cases by September 2015, possibly giving access to the first testers by June 2015.
A fully production-level infrastructure might be ready in the early 2016, provided adequate human and financial resources will be available.