DOI: 10.26314/GARR-Conf16-proceedings-15

# RMLab: Gestione Agile di un data center distribuito

Antonio Budano[1], Federico Zani[2]

*[1]INFN Sezione di Roma Tre, [2]INFN Sezione di Tor Vergata*

**Abstract.** Il progetto RMLab nasce nel 2015 grazie alla collaborazione delle sezioni INFN di Roma Tor Vergata, Roma Tre e dei Laboratori Nazionali Di Frascati, con il supporto fondamentale del GARR per il networking. L'idea è nata dall'esigenza di voler creare un'infrastruttura distribuita che fornisse una piattaforma di Cloud Computing di tipo Iass basata su Openstack, mettendo in comune le competenze e le risorse di diverse sezioni.

**Keywords.** Cloud Computing, Openstack, Data center distribuito, Automazione sistemi

## Introduzione

Il progetto RMLab è iniziato nel 2015 grazie alla collaborazione delle sezioni di Roma Tor Vergata, Roma Tre e Laboratori Nazionali di Frascati dell'INFN, con il supporto fondamentale del GARR. L'idea è nata dalla necessità di creare una infrastruttura distribuita per la costruzione di una piattaforma di Cloud Computing di tipo Iaas basata su OpenStack, mettendo in comune le competenze e le risorse dei diversi data center.

Alla base di questa soluzione vi era la necessità di avere un livello di rete che permetterebbe tre centri dell'INFN di comunicare di in modo sicuro, trasparente ed efficiente. La soluzione tecnica offerta dal GARR per la realizzazione di questo strato è stato quello di creare tre reti private per comunicare isolare il traffico dal resto delle reti interne ed esterne.

Per gestire questo tipo di piattaforma, è stato necessario definire un nuovo modello di gestione di data center distribuiti. Questo modello consente una gestione agile dell'intero sistema in modo trasparente, rimuovendo i confini fisici del posizionamento hardware. L'attuazione dei vari servizi è stata effettuata su tre diverse tecnologie: servizi sono stati distribuiti su macchine fisici, macchine virtuali e container.

L'adozione di sistemi di automazione "IT" sono stati fondamentali per la gestione di questa infrastruttura. Abbiamo, infatti, implementato i servizi che ci hanno permesso di semplificare le installazioni, le configurazioni e gli aggiornamenti, utilizzando prodotti open source come Foreman, Puppet e Docker. Inoltre, un componente chiave per la gestione dell'infrastruttura è sicuramente il sistema di monitoraggio e di allarme, in grado di segnalare rapidamente ed efficacemente ogni malfunzionamento del sistema.

## 1. L'infrastruttura e il modello di rete

L'infrastruttura del progetto RMLab è distribuita sui tre data center presenti all'interno delle strutture INFN sfruttando le risorse informatiche locali.

L'idea principale del progetto era di avere un'unica infrastruttura gestita in maniera trasparente da ogni amministratore in qualunque sede. La realizzazione di tale infrastruttura richiede di avere una connessione sicura, trasparente ed efficiente. GARR ha quindi elaborato una soluzione basata sull'instradamento di tre reti private (uno per ogni sito) utilizzando il protocollo "Label Switching Multi-Protocol" (MPLS). Una figura schematica di questa architettura è mostrata in Figura 1. In particolare il GARR ha creato un Layer 3 Virtual Private Network, con regole di routing virtuali (VRF – Virtual Routing and Forwarding) su ogni router presente nelle tre sedi INFN.

I vantaggi di questa soluzione sono principalmente: la Quality of Service (QoS), le elevate prestazioni e l'isolamento del traffico (garantita da GARR). I tre siti, sono collegati tra loro

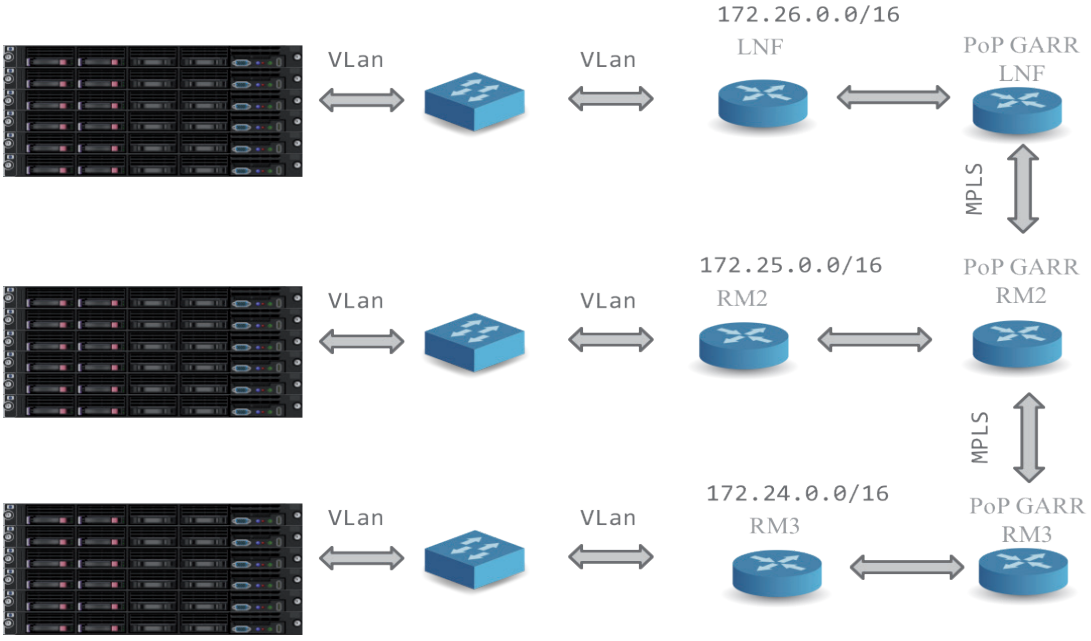



**Fig. 1** Schematizzazione del modello di rete dell'infrastruttura RMLab

con una connessione ad 1 Gbps, nella tabella 1 riportiamo i risultati di alcuni test in cui è possibile osservare la bassa latenza della connessione, misurata utilizzando il comando ping mentre nella tabella 2 riportiamo la larghezza di banda, misurata con l'applicazione iperf.

| | ROMA 2 | ROMA 3 | LNF |
|---|---|---|---|
| **ROMA 2** | --- | 1,12 | 1,24 |
| **ROMA 3** | 1,09 | --- | 1,32 |
| **LNF** | 1,26 | 1,31 | --- |

*Fonte: misurazione mediata su due server all'interno di ogni rete -2016*

**Tab. 1** Latenza (ms)

| | ROMA 2 | ROMA 3 | LNF |
|---|---|---|---|
| **ROMA 2** | --- | 916 | 921 |
| **ROMA 3** | 916 | --- | 903 |
| **LNF** | 920 | 905 | --- |

*Fonte: misurazione mediata su due server all'interno di ogni rete -2016*

**Tab. 2** Larghezza di banda (MB/s)

Nella figura 2 è riportato uno schema dei servizi principali necessari al funzionamento e al monitoraggio dell'intera infrastruttura. L'implementazione dei vari servizi è stata effettuata su tre diverse tecnologie, non legandoci mai ad una unica soluzione predefinita: abbiamo servizi su macchine fisiche, su macchine virtuali e su Docker container, considerando volta per volta la tecnologia migliore rispetto alle caratteristiche del servizio che andava implementato.

La ridondanza e la locazione di ogni servizio è stata scelta in modo da garantire il funzionamento del sistema in caso di guasto di un intero sito. Come si può osservare nella figura 2, il sistema ha un servizio DNS (Domain Name Server) per il dominio interno, replicato su tutti i data center aggiornabile in autonomia dai sistemi stessi tramite il comando nsupdate. Questo permette di cambiare velocemente la disposizione dei servizi a livello geografico senza che l'infrastruttura ne subisca conseguenze. Il DNS viene usato anche per gestire l'alta affidabilità (High Availability) di alcuni servizi tramite script che vanno a modificare i record in round robin, dopo aver controllato quali degli endpoint sono attivi. L'idea principale, è che il servizio possa essere spostato il più rapidamente possibile da una sede all'altra, riducendo al minimo il downtime. Ogni sito ha inoltre un proprio server NAT (Network Address Translation) per permettere ad ogni macchine di poter accedere alla rete In-

ternet, principalmente per i processi di installazione ed aggiornamento software.

È presente inoltre un Percona XtraDB Cluster necessario a garantire la replica di un database MySQL comune a tutta l'infrastruttura e due file system: uno di tipo posix basato su IBM General Parallel File System (GPFS) e uno di tipo object storage basato su CEPH, entrambi replicati su ogni sede. L'accesso all'infrastruttura (per gli utenti e I servizi) è garantita dal sistema di autenticazione nazionale INFN AAI.

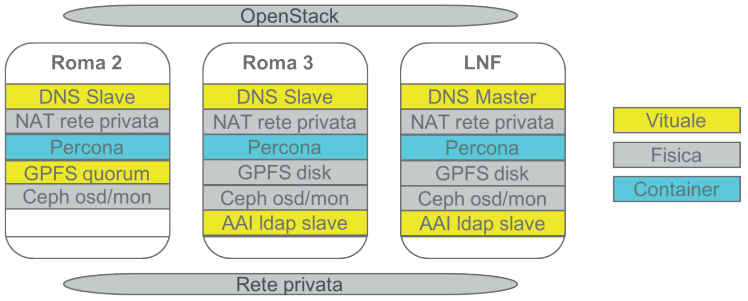



**Fig. 2** Schematizzazione dei servizi principali al funzionamento e al monitoraggio dell'infrastruttura

L'infrastruttura cosi costruita ci è servita per la creazione di un una Cloud privata distribuita di tipo IaaS basata su Openstack (Mitaka release) dove ogni sito viene gestito come una availability zone. In questo momento l'infrastruttura offre circa 350 core, 750GB di memoria RAM e circa 5TB di spazio disco locale. Il cluster CEPH offre circa 36TB (replica 3).

## 2. Modello di gestione

Le risorse messe a disposizione dell'infrastruttura, distribuite sui diversi data center, sono molto eterogenee tra di loro. Questo ha reso necessario, fin dall'inizio della messa in opera del sistema, un'analisi del modello di gestione migliore che permettesse di poter operare su ogni componete dell'infrastruttura in maniera semplice, snella e trasparente rispetto ad ogni componete del gruppo di lavoro.

La soluzione adottata si basa su due prodotti open-source Foreman e Puppet. Il primo permette, in maniera automatizzata, di gestire l'installazione di server (fisici o virtuali) tramite server pxe/dhcp: I limiti derivanti dall'avere 3 domini di broadcast separati, che impedirebbero il funzionamento di questi protocolli, vengono superati grazie all'utilizzo di dhcp relay presenti in ognuno dei 3 router di frontiera.

Il secondo permette di gestire in maniera automatizzata le configurazioni dei vari servizi presenti su ogni componente dell'infrastruttura. Foreman include, inoltre, un ENC (external node classifier) per Puppet, per la gestione di una macchina in base ad un determinato profilo: dall'installazione fino alla configurazione completa. Con questo sistema è possibile implementare facilmente e automaticamente qualsiasi server nell'infrastruttura (indipendentemente dalla sua locazione), replicare installazioni, monitorare e gestire l'infrastruttura, automatizzare le operazioni ripetitive, distribuire rapidamente nuove applicazioni e gestire in modo rapido le configurazioni. Nella figura 3 viene riportato schematicamente un esempio di come avviene la fase di installazione e configurazione di un qualsiasi componente all'interno dell'infrastruttura.

Un ulteriore vantaggio nel gestire l'infrastruttura in questo modo è quello di poter realizzare ogni servizio attraverso una sequenza di comandi e quindi definire attraverso diversi codici lo stato dell'intera infrastruttura. Per la gestione di tutto il software dell'infrastruttura abbiamo adottato un sistema di archiviazione e versioning del codice basato su Git e tutta la documentazione viene riportata su una Wiki privata.

Di fondamentale importanza nella gestione dell'intera infrastruttura sono i servizi di monitoraggio, allarmistica e diagnostica. Il sistema di monitoraggio e allarmistica è basato su Zabbix, che permette di avere lo stato di ogni macchina e, attraverso dei plugin dedicati, lo stato dei servizi, notificando ogni problema via e-mail o sms. Il servizio di diagnostica, l'analisi dei failure e dei log e le verifica delle metriche vengono eseguite attraverso ELK (Elasticsearch, Logstash, and Kibana).

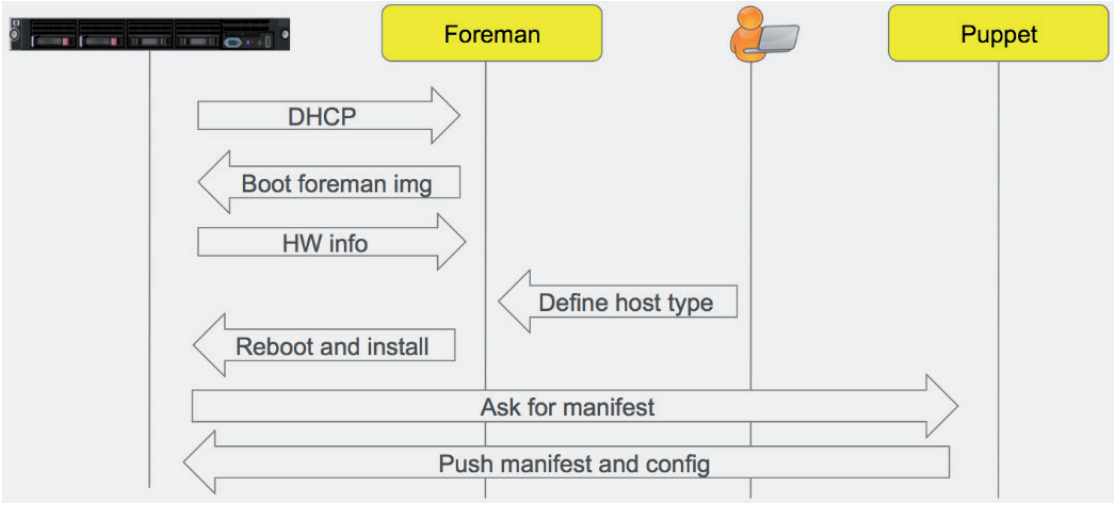



**Fig. 3** Schematizzazione della fase di provisioning di Foreman/Puppet.

All'infrastruttura abbiamo aggiunto un ulteriore sistema di notifiche basato sul sistema HipChat. Su tale sistema vengono collezionate le notifiche provenienti dal sistema di monitoraggio e allarmistica, le notifiche provenienti dall'infrastruttura Openstack e le comunicazioni degli amministratori. Questo sistema permette ad ogni amministratore di vedere lo stato dell'infrastruttura e notificare gli altri colleghi delle operazioni che si stanno effettuando. Nella figura 4 è riportata una schermata di HipChat dove si posso osservare I messaggi provenienti dal sistema di monitoraggio e dall'infrastruttura Openstack.

## 3. Conclusioni

Grazie alla soluzione offerta da Garr abbiamo potuto connettere i tre siti INFN (i Laboratori Nazionali di Frascati, Roma Tor Vergata e Roma Tre) del progetto RMLab in maniera sicura, trasparente ed efficiente permettendoci di poter creare un'infrastruttura affidabile. L'implementazione di servizi di automazione ci ha permesso di creare e mantenere efficacemente l'intero sistema anche con risorse eterogenee tra loro e localizzate in sedi differenti. Il sistema di monitoraggio, allarmistica e diagnostica permettono a ciascun componente del gruppo di amministratori di poter intervenire su ogni componente del

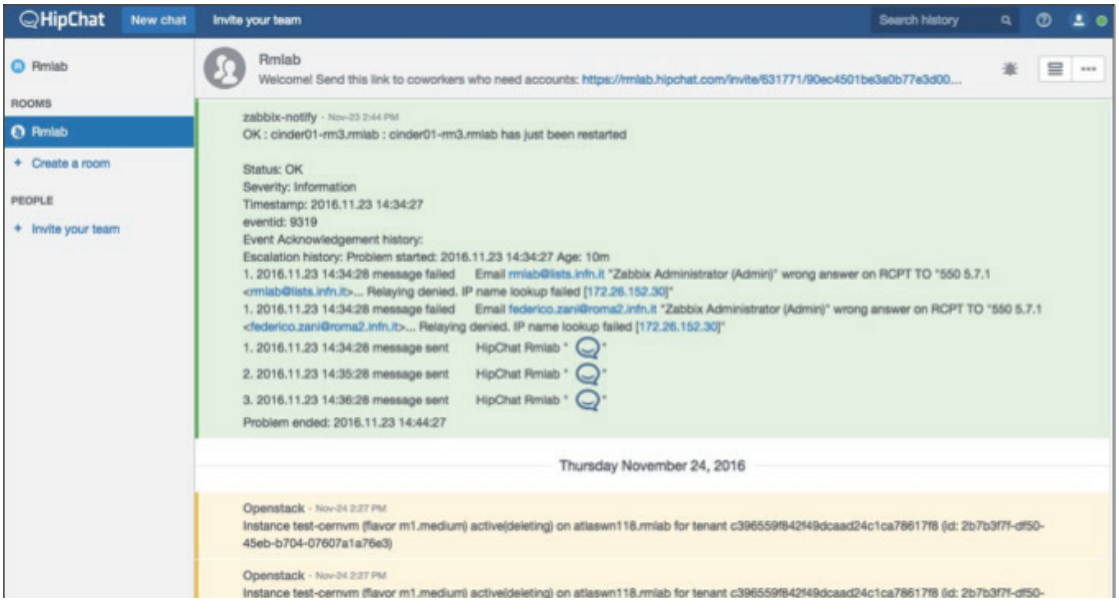


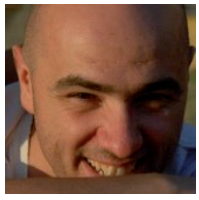

**Fig. 4** Schermata esempio del sistema HipChat.

sistema assicurando una gestione efficiente di tutta l'infrastruttura. L'esperienza nella creazione dell'infrastruttura RMLab ci ha permesso di consolidare le nostre competenze nella gestione dei data center. Grazie soprattutto alla collaborazione che abbiamo instaurato e alla diversità di competenze, abbiamo imparare molto gli uni dagli altri. Nel prossimo futuro, grazie al supporto del Garr, contiamo di avere un aumento della banda di connessione a 10 Gbps tra le tre sedi, questo ci permetterà di avere un sistema molto più efficiente.

Antonio Budano
antonio.budano@roma3.infn.it

Dal 2005 lavora presso l'INFN dove si occupa della gestione dei servizi informatici e del cluster di calcolo della sezione di Roma Tre. Durante la sua esperienza nell'INFN si è occupato di sistemi di acquisizione e trasferimento dati degli esperimenti Argo-YBJ e KLOE e delle architetture di computing di tipo grid e cloud.

Federico Zani
federico.zani@roma2.infn.it



Da quasi 15 anni si occupa di sviluppo software e amministrazione di sistemi unix con una attenzione particolare ai sistemi distribuiti e alla loro scalabilità, la cui naturale evoluzione è verso tecnologie cloud. Come staff INFN si occupa sia di progetti di rilevanza nazionale che di gestire l'infrastruttura locale di Roma Tor Vergata, dando anche supporto allo sviluppo software.