

**OPEN ACCESS**

# Testing an Open Source installation and server provisioning tool for the INFN CNAF Tier1 Storage system

To cite this article: M Pezzi *et al* 2014 *J. Phys.: Conf. Ser.* **513** 032075

View the [article online](#) for updates and enhancements.

## Related content

- [Upgrade and integration of the configuration and monitoring tools for the ATLAS Online farm](#)  
S Ballestrero, F Brasolin, G-L Darlea *et al.*
- [Integrating configuration workflows with project management system](#)  
Dimitri Nilsen and Pavel Weber
- [A Voyage to Arcturus: A model for automated management of a WLCG Tier-2 facility](#)  
Gareth Roy, David Crooks, Lena Mertens *et al.*

# Testing an Open Source installation and server provisioning tool for the INFN CNAF Tier1 Storage system

M Pezzi<sup>1</sup>, M Favaro<sup>1</sup>, D Gregori<sup>1</sup>, PP Ricci<sup>1</sup>, V Sapunenko<sup>1</sup>

<sup>1</sup> INFN CNAF Viale Berti Pichat 6/2 40127 Bologna, Italy

E-mail: [michele.pezzi@cnafe.infn.it](mailto:michele.pezzi@cnafe.infn.it)

**Abstract.** In large computing centers, such as the INFN CNAF Tier1 [1], is essential to be able to configure all the machines, depending on use, in an automated way. For several years at the Tier1 has been used Quattor[2], a server provisioning tool, which is currently used in production. Nevertheless we have recently started a comparison study involving other tools able to provide specific server installation and configuration features and also offer a proper full customizable solution as an alternative to Quattor. Our choice at the moment fell on integration between two tools: Cobbler [3] for the installation phase and Puppet [4] for the server provisioning and management operation. The tool should provide the following properties in order to replicate and gradually improve the current system features: implement a system check for storage specific constraints such as kernel modules black list at boot time to avoid undesired SAN (*Storage Area Network*) access during disk partitioning; a simple and effective mechanism for kernel upgrade and downgrade; the ability of setting package provider using yum, rpm or apt; easy to use Virtual Machine installation support including bonding and specific Ethernet configuration; scalability for managing thousands of nodes and parallel installations. This paper describes the results of the comparison and the tests carried out to verify the requirements and the new system suitability in the INFN-T1 environment.

## 1. Introduction

The INFN CNAF computing center hosts the Tier1 site [1], the largest Italian computing facility used in the LHC distributed computing infrastructure, equipped with 120 racks in an area of 1000 m<sup>2</sup>. In a computing center like Tier1, is essential to be able to configure all the machines, depending on their use, in an automated way. For several years at the Tier1 we have been using Quattor, a system administration toolkit for the automated installation, configuration, and management of clusters and farms. Quattor [2] is developed as a community effort and provided as open-source software, and it was originally developed to manage the clusters and services used in grid computing. Even if Quattor is currently used in production at INFN CNAF efficiently, we have recently started a comparison study involving other tools able to provide specific server installation and configuration features and also to offer a proper full customizable solution as an alternative to Quattor. The study resulted in the adoption of two tools: Cobbler [3] and Puppet [4]. Cobbler is a Linux installation service and Puppet is a software which allows automated and centralized management of an infrastructure of Linux and general Unix systems; both are open source software.

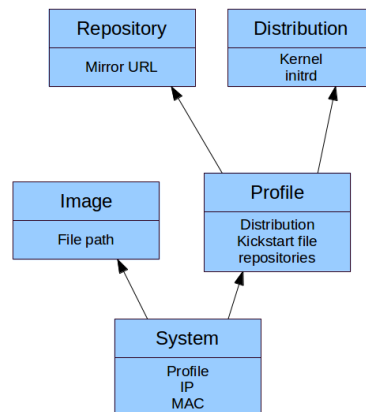


## 2. Cobbler and Puppet : how they works together

Cobbler [3] is a Linux installation service that allows for rapid setup of network installation environments. It automates many associated Linux tasks so there is no need to hop between various commands and applications when deploying new systems or changing existing ones. Cobbler can help with network provisioning, managing DNS and DHCP services, and also provides a tool, “Koan”, for simplifying virtualization deployments. Koan is a client-side helper program which allows for both network provisioning of new virtualized guests and reinstallation of an existing system. When invoked, Koan requests install information from a remote Cobbler boot server, then it “kicks” off installations based on what is retrieved from Cobbler and what is specified on the Koan command line. This program can be installed automatically by Cobbler.

The configuration structure of Cobbler is based on a set of registered objects, each of which represents an entity that is associated with another entity. When one object points to another object, it inherits the data and can override or add more specific information. The following types of objects (Figure 1) are defined:

- *Distribution*: Represents an operating system. It carries information about the kernel and initrd, plus other data such as kernel parameters.
- *Profile*: Points to a distribution, a kickstart file, and possibly repositories, plus other data such as more specific kernel parameters.
- *System*: Represents the “machine” to be provisioned. It points to a profile or an image and contains information about IP and MAC addresses, power management (address, credentials, type), and more specialized data.
- *Repository*: Holds mirroring information for a yum or rsync repository.
- *Image*: Can replace a distribution object for files that do not fit in this category (for example, cannot be divided in kernel and initrd).



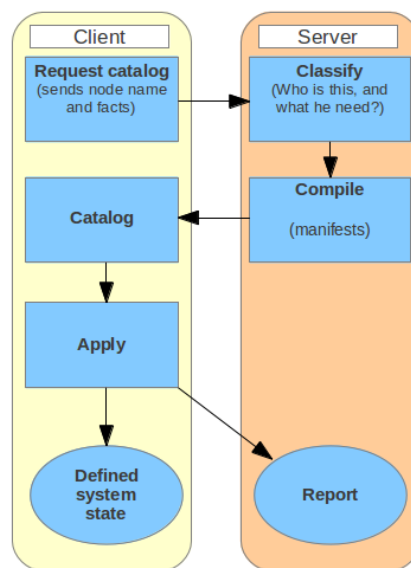
**Figure 1.** Cobbler objects relationship.

According to the objects registration and the association between them, Cobbler knows how to change the file system to reflect the configuration.

Puppet [4] is an open source software, written in Ruby, which allows automated and centralized management of an infrastructure of Linux and Unix systems. Puppet is a client-server system, the client (`puppetd`) is launched (such as single command or service) on the system to set up and it connects to the server (the Puppet Master) through TCP port from

which retrieves the profile configurations associated with its hostname. The network traffic is encrypted, following an exchange of certificates between client and server that guarantees the authenticity and integrity of the communication. The profiles of the machines are defined within files called “*manifest*” which contains all the resources that are associated with the managed host. All the manifests are stored in the Puppet Master.

The logic of Puppet is to define the state of a system and every time the client runs it make sure that the state is applied to client itself. The first time `puppetd` run, it installs all the packages, it starts services and it changes configuration file as defined in the Puppet Master. All subsequent times, in theory, if there is no changes in the *manifest* the `puppetd` should not change anything on the client.



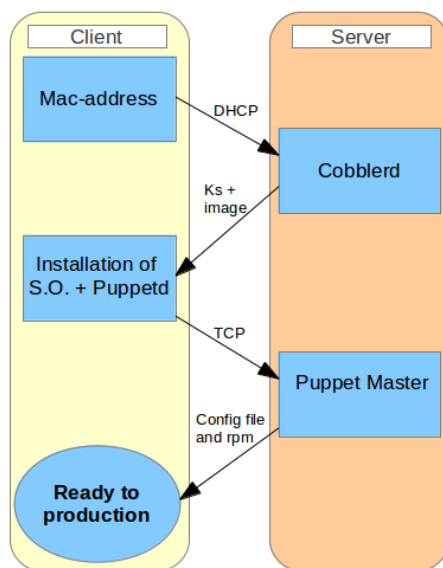
**Figure 2.** How Puppet works: the interaction between client and server

In Figure 2 is shown, in simplified way, the operation of Puppet:

- 1- Whenever a client node connects to the Puppet Master, the master server analyzes the configuration (the *manifest* file) to be applied to the node, and how to apply the configuration on the node.
- 2- Puppet Master server takes and collects all the resources and configurations to be applied to the node, and compiles it and make it a catalog. This catalog is given to the puppet agent of the node.
- 3- Puppet agent will apply the configuration on the node, according to the catalog, and then reply back, and submit the report of the configuration applied to the puppet master server.

The two software tools can work together. Cobbler during post-install phase automatically installs Puppet. Cobbler also manages the Puppet certificates: it removes old certificate of the client and registers the new certificate when Puppet client requests the sign of the certificate. In order for the two services to work together they must be provided by a single server machine.

Figure 3 summarizes the operation of the two software tools when they work together. When a new machine is booting, the PXE boot ROM sends a DHCP request on the network. The DHCP-service, managed by Cobbler, responds to the request providing ip-address and all the



**Figure 3.** How Puppet and Cobbler works together.

necessary to install the machine. During the post-install phase Puppet is installed. At the end of the installation process the Puppet client contacts the Puppet Master to download the configuration and applies it. From this point onwards the client contacts the server just to keep updating the system state.

### 3. Functionality tests

Tests were performed with two different test-benches. The first, called Testbed [6] consists of 15 machines on the same network with different hardware, is used to test the new software, hardware, or upgrade procedures. The second, called OpenLab, is a testing facility with its own infrastructure dedicated for exploration of “state-of-art” hardware and software solutions in the field of scientific computing. To perform this exercise we have used 1 server, a bastion-host running proxy and providing access and 32 computing nodes. In both cases we have used the two software packages coupled into a single system, to install and configure all the machines. In the Testbed we install the machines with different operating system and different kernel version. To install Puppet we also defined in Cobbler profile an external repository with Puppet packages (EPEL) and we locally mirrored the repository (since the mirror can be done directly by Cobbler). We did several tests to compare Cobbler-Puppet system to Quattor. In particular we tested the following features:

- *Black list of kernel modules at boot.* It’s essential to blacklist fibre channel module at boot to prevent access to the SAN during the installation phase. In this way we can allow access to the SAN only when required. In the Cobbler profiles is already implemented the option to blacklist modules at boot time, so it is sufficient to properly edit the profiles.
- *Kernel upgrade and downgrade.* For this operation it is preferred to use Puppet: we can handle the rpm packages using a function already implemented within Puppet which allows to install packages by some constraints.
- *Setting package providers.* Puppet has a function to change the package provider like yum, rpm or apt.

- *Configuration of the network.* The network configuration is set, unless explicitly requested, through Cobbler. In fact it is possible to define bridges for virtual machines, bonding or single interfaces. For the configuration of bonding we use Puppet to have more control over the configuration file.
- *Installation of virtual machines.* To install the virtual machines using Cobbler the additional software “Koan” should be used to install the virtual machines using Cobbler; it must be installed on the guest machine

In the OpenLab we installed all the machines with the same operative system and same kernel. In this test-bench we decided to not mirror the rpm repository and use an external one. To reach the external repository we defined the ip-address of proxy-server within the Cobbler profile. Actually the Cobbler-Puppet system is used to maintain two type of machine configuration: the first is for a MySQL tests and the second for GPFS [7] tests.

During the test phase, we are was able to achieve the following results:

- We are able to integrate the two software and we managed to get them to work together.
- We have created modules to manage: the network interfaces in bonding, the access to the SAN, the kernel and other packages in a dinamic way; for example, for configuring bonding the configuration depends on some constraints like the number of network interface.
- We could opt to the package provider (yum, rpm or apt).
- The installation of a machines requires on average 10 minutes and the same time is required for a parallel installation of 10 machines.
- Cobbler is able to install virtual machines via Koan: the profiles of the virtual machines are managed in the same manner as the real ones.

#### 4. Conclusion

The tests we performed showed that the Cobbler and Puppet system has been able to fulfill the constraints we had set for the systems management and they can be used in production environment of the INFN CNAF Tier1 as installation and configuration tools. The Cobbler and Puppet system can be a good alternative to Quattor. The next tests will include a scalability challenge in order to install and configure hundreds of nodes i.e. a Tier1-size computing farm. At present the system is still used to manage the CNAF Tier1 Openlab machines.

#### References

- [1] G. Bortolotti et. al Proceedings of CHEP 2012, *The INFN Tier-1*, Computing in High Energy and Nuclear Physics 2012 New York Journal of Physics: Conference Series Volume 396 Part 4
- [2] *Quattor*, <http://quattor.org/>
- [3] *Cobbler*, <http://www.cobblerd.org>
- [4] *Puppetlabs*, <http://www.puppetlabs.com>
- [5] Paulo de Rezende Pinatti, *Automate and manage systems installation with Cobbler*, Staff Software Engineer, IBM
- [6] D. Gregori et Al., *INFN Tier-1 Testbed Facility*, 2012 J. Phys.: Conf. Ser. 396 042024
- [7] *GPFS*, <http://www-03.ibm.com/systems/software/gpfs/>