PROCEEDINGS OF SCIENCE

A Web-based portal to access and manage WNoDeS Virtualized Cloud resources

Davide Salomoni[*] **INFN-CNAF**
E-mail: davide.salomoni@cnaf.infn.it

Daniele Andreotti INFN-CNAF
E-mail: daniele.andreotti@cnaf.infn.it

Luca Cestari University of Ferrara
E-mail: cestari.luca@gmail.com

Guido Potena University of Ferrara
E-mail: guido.potena@student.unife.it

Peter Solagna INFN Padova[†]
E-mail: peter.solagna@pd.infn.it

The Worker Nodes on Demand Service (WNoDeS), developed by INFN, is a framework designed to offer local, grid or cloud-based access to computing and storage resources, preserving maximum compatibility with existing computing center policies and workflows. WNoDeS has been running in production at the INFN Tier-1 located at CNAF since November 2009, where it currently manages several thousands of dynamically created Virtual Machines; WNoDeS is also being deployed at several other Italian sites. This presentation shows the current state of the WNoDeS Web portal, aimed at end users for the instantiation of WNoDeS cloud resources and at site administrators for graphical management of the WNoDeS infrastructure. The portal is scheduled to be released in the next major version of WNoDeS (WNoDeS 2), expected by Fall 2011.

The International Symposium on Grids and Clouds and the Open Grid Forum - ISGC2011,
March 25-30, 2011
Academia Sinica, Taipei, Taiwan

[*] Speaker.
[†] Now at EGI.eu, Amsterdam.

1. Introduction

The INFN WNoDeS [1] (Worker Nodes on Demand Service) is a virtualization architecture targeted at Grid/Cloud integration. It provides transparent user interfaces for Grid, Cloud and local access to resources, re-using several existing and proven software components like Grid authentication and authorization mechanisms, KVM-based virtualization, local accounting, monitoring and work-flows, and data center schedulers. WNoDeS is in production at the INFN Tier-1 [2], Bologna, Italy since November 2009. Several million production jobs, including those submitted by experiments running at the LHC [3], have been processed by WNoDeS. Currently, about 2,000 Virtual Machines (VMs) are dynamically created and managed by WNoDeS within the INFN Tier-1 computing infrastructure. Recently, WNoDeS has been installed by an Italian WLCG Tier-2 site, with other domestic and international sites considering its adoption.

WNoDeS uses Linux KVM [4] to virtualize resources on-demand; the resources are available and customized for:

- direct job submissions by local users,
- Grid job submissions (with direct support for the EMI CREAM-CE and WMS components),
- instantiation of Cloud resources or instantiation of Virtual Interactive Pools (VIP)

VM scheduling is handled by a LRMS (a “batch system software”): there is no need to develop special (and possibly unscalable, inefficient) resource brokering systems. The LRMS is totally invisible to users for e.g. Cloud instantiations; in particular, there is no concept of a “Cloud over Grid” or “Grid over Cloud” hierarchy: WNoDeS simply sees and uses all resources, dynamically presenting them to users as users want to see and access them.

This paper describes how Cloud resources may be self-provisioned by users through a web-based portal. The same portal may also be used by WNoDeS system administrators to monitor usage of WNoDeS-based resources, without resorting to command-line tools.

2. The WNoDeS Cloud Portal

The current WNoDeS Web portal has three main components:

- The Authentication and authorization module.
- The Cloud request module.
- The WNoDeS cloud management module.

These three modules are integrated into a Web-based application, called the WNoDeS Cloud Portal. The following paragraphs briefly describe each of these modules. A subsequent section will then describe some aspects related to the implementation of the portal.

2.1 The Authentication and authorization module

This module lets existing Grid users connect to the portal using a valid X.509 digital certificate installed in a user's browser. Before issuing any requests to the portal, the user must specify the existing Grid VO he intends to charge the requests to. The *authentication* part of the module then contacts the appropriate VOMS [5] server(s) for the specified VO and validates whether the user is actually part of that VO. If this check is successful, the *authorization* part of the module contacts an Argus [6] server to verify the authorization policies that have been either implicitly or explicitly defined for the user. These policies may include whitelists or blacklists, DN- or role-based access, or other types of Argus-supported policies.

2.2 The Cloud request module

This module lets users select, instantiate and manage computing and storage resources. These resources are categorized for ease of management and for billing purposes. For example, a user may wish to instantiate a so-called "Small server": as detailed later on, this results in the request for a virtualized system with one core, 1.7 GB RAM, and 50 GB of local hard disk space. This server is dynamically created by the WNoDeS system using the WNoDeS common pool of resources and handed over to the user, who can connect to it using a passwordless `ssh` connection. Wallclock time usage of the resources is billed to the user or VO requesting it. The user can eventually destroy the resource once he is done with it.

2.3 The WNoDeS cloud management module

A WNoDeS administrator may log on to the portal and see running VM handled by the WNoDeS system. In particular, the state of the WNoDeS Name Server and of all the WNoDeS Virtual Machines may be examined. Through an interface with the KVM `libvirt` virtualization API, statistics on the use of the WNoDeS VMs like CPU, RAM, or I/O usage can be shown. This module also provides the possibility to start and stop WNoDeS services.

3. Virtual Machine Definition

The WNoDeS architecture, showed in Fig. 1, allows specification of all the parameters defining a VM. Preliminary versions of the Cloud portal fully exploited this flexibility. However, letting users handle too many variables for the definition of Cloud resources results in confusion to the users themselves and into serious additional burden to resource providers without apparent benefits. Early experience with the provisioning of cloud resources at the INFN Tier-1 showed that most customer requests can instead be satisfied with a fixed set of parameters characterizing the VMs.

In particular, for Cloud requests we defined the following standard instance types:

- Small : 1 core, 1.7 GB RAM, 50 GB local HD
- Medium : 2 cores, 3.5 GB RAM, 100 GB local HD
- Large : 4 cores, 7 GB RAM, 200 GB local HD
- Extra-large: 8 cores, 14 GB RAM, 400 GB local HD

The amount of RAM for each instance is defined taking into account the possible overhead caused by the Operating System. In practice, this overhead may be reduced, thanks for example to modern features of the Linux operating system like Kernel Samepage Merging [12]. Depending on usage patterns and hardware capabilities, therefore, the parameters above may be appropriately redefined.

The billing model employed for WNoDeS naturally reflects the resource parametrization defined above. Since WNoDeS is closely coupled with a Local Resource Management System (LRMS) for the provisioning of resources, accounting and billing are based on the LRMS itself, without the need for additional components.

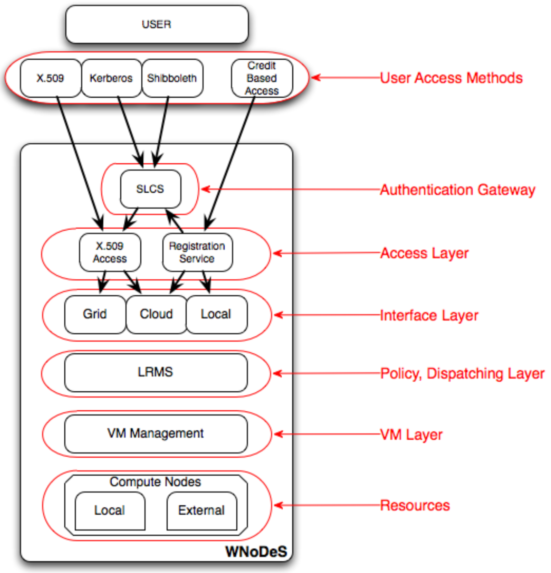


Figure 1: The general architecture of WNoDeS.

Custom requests requiring different parameters for VMs may still be handled, with bespoke solutions that can be discussed and arranged between customers and resource providers. In general, though, it is to be noted that, in addition to those shown above, several other parameters may be used to characterize VMs. For example, customers often request the definition and use of *custom images*; these will be implemented in WNoDeS through modification of pre-defined VM image sandboxes and their subsequent saving and retrieval, or the association of permanent storage to VMs, possibly with clearly defined service characteristics. In particular, in a future WNoDeS release we plan to support requests for so-called *whole-nodes* type of resources, characterized as follows:

- whole-node, *hard* type: this is a request where the customer requires *all* cores found on a hardware platform, (1.7 * num. cores) GB RAM, (50 * num. cores) GB local HD
- whole-node, *soft* type: the customer requires all *available* cores (possibly with a specified minimum), (1.7 * num. cores) GB RAM, (50 * num. cores) GB local HD

The latter type of *whole-node* resources are normally much more easily provisioned than the former ones, since the typical advance reservation mechanisms (which in WNoDeS are easily supported, thanks again to its strong coupling with a LRMS) applied to reserve cores do not require in that case physical hardware to be completely job-free before the request can be satisfied.

Once a given instance type is selected, the user is requested to select an Operating System to be installed on it. We currently only support Linux, although preliminary testing of VM instances running Microsoft Server 2008 with HPC Pack are ongoing.

4. The Cloud Web Portal

WNoDeS has been supporting the OGF Open Cloud Computing Interface (OCCI) [7] since

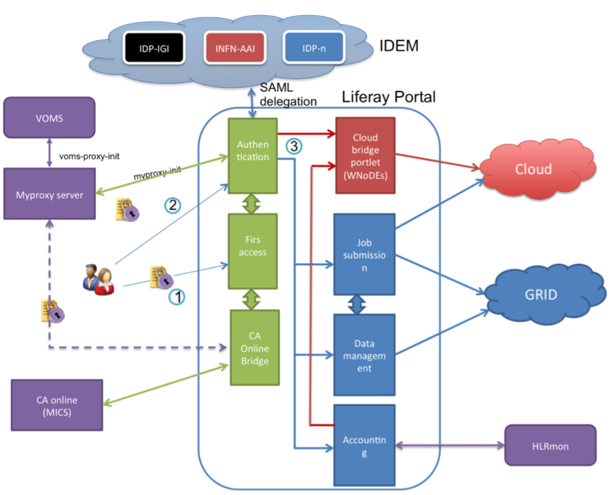


Figure 2: WNoDeS as part of a generic submission portal.
(picture courtesy M. Bencivenni, INFN-CNAF)

2009. However, direct access to the OCCI layer is uncommon. For all practical purposes, the instantiation of Cloud resources in WNoDeS is therefore realized through the Cloud Web Portal, which can be seen as a user-friendly interface to the OCCI layer.

It is important to note, though, that WNoDeS is strictly speaking neither a Cloud nor a Grid provisioning system. As it should be clear from the architectural diagram in Fig. 1, WNoDeS is a generic system for the scalable virtualization of *resources* and for their presentation to users through multiple interfaces, be they called Grid, Cloud, or else. For this reason, we defined an architecture, shown in Fig. 2, for a generic submission portal, of which the WNoDeS Cloud portal is only a subset. The complete portal, which is currently being developed, integrates authentication, authorization and accounting for both Grid and Cloud type of resource requests.

4.1 Process flow for WNoDeS Web-based Cloud VM instantiation

The web application implementing the instantiation of Cloud VMs follows a traditional MVC (Model-View-Controller) programming model. This is realized through the Spring [8] MVC Java framework, where the *Model* part is coordinated with Hibernate [9], the *view* part is implemented with FreeMarker [10] and the *controller* is the application core logic, handled in Java.

Successful connection to the Tomcat-based Web application currently requires the presence of an X.509 digital certificate in the user's browser. For the user to be able to instantiate Cloud resources, he must be part of an existing Grid Virtual Organization (VO).[1] The current supported use case is therefore one of serving existing Grid users wishing to instantiate Cloud resources *a)* charging these instantiation to their VO, and *b)* without requiring the users to acquire new credentials.

The full program flow for Cloud VM instantiation is shown in Fig. 3. In practice, the user needs to follow four steps to instantiate resources, described below.

[1] This is not a strict requirement of the portal architecture. The generic portal referred to above is being developed with the requirement in mind, that users authenticated through X.509 *and* other methods must be able to self-provision and use Grid or Cloud resources.

4.1.1 Authentication, authorization and selection of resource type and operating system

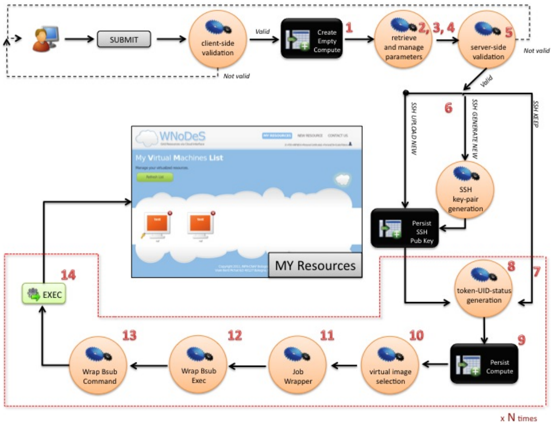
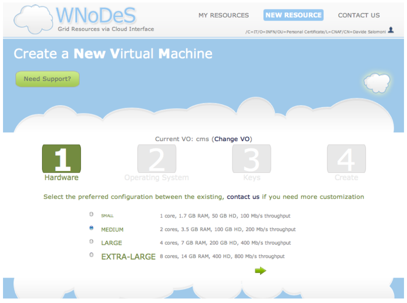


Figure 3: Cloud instantiation process flow.

Since a generic X.509 certificate like the one users have in their browser does not contain VO-related info, a user is requested to manually select a VO among the ones supported by the WNoDeS portal installation. At this stage, a VOMS validation step is performed, making sure the user is actually part of the VO he selected. Success in this validation phase is needed to proceed further on. From an implementation point of view, the validation is realized through queries made to the VOMS servers for the supported VOs.

After successful VOMS authentication is performed, a call is made to an Argus server to verify whether the user is authorized to perform instantiation of Cloud resource. The policies in Argus may range from very simple to rather complex ones; for example, it is possible to authorize users based on whitelists, on roles present in the user's DN, and so on.

If the authorization process is successful, the user may actually in Step 1 select a Cloud resource type. An example of this selection is shown in Fig. 4.



/C=IT/O=INFN/OU=Personal Certificate/L=CNAF/CN=Davide Salomoni

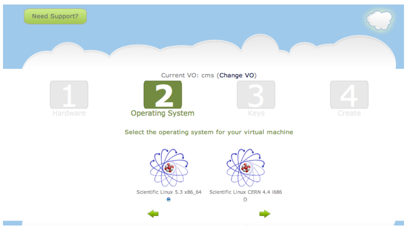
Create a **New Virtual Machine**

Current VO: cms (Change VO)

Select the preferred configuration between the existing, contact us if you need more customization

○	SMALL	1 core, 1.7 GB RAM, 50 GB HD, 100 Mb/s throughput
●	MEDIUM	2 cores, 3.5 GB RAM, 100 GB HD, 200 Mb/s throughput
○	LARGE	4 cores, 7 GB RAM, 200 GB HD, 400 Mb/s throughput
○	EXTRA-LARGE	8 cores, 14 GB RAM, 400 HD, 800 Mb/s throughput

Figure 4: Resource type selection.



Current VO: cms (Change VO)

Select the operating system for your virtual machine

● Scientific Linux 5.3 x86_64 ○ Scientific Linux CERN 4.4 i686

Figure 5: Operating system selection.

After the resource type is selected, in Step 2 the user can choose the Operating System to be installed on the resource type, as shown in Fig. 5.

4.1.2 Access to the virtualized resources

Access to the created virtualized resources is realized through passwordless `ssh`. Step 3 of the Web application supports two option to use `ssh`:

- Users who already have their own private/public `ssh` keypair or know how to generate one may simply upload their public key to the Web application.

- Users who do not know how to generate a private/public `ssh` keypair or who for any reason do not want to upload their public key to the Web application may choose to let it generate a keypair for them.

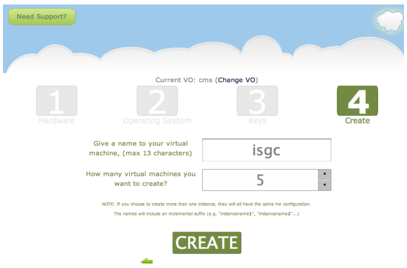
The latter case obviously implies a certain degree of trust that the Web application, the system the application is running on, WNoDeS itself and the resource provider will not improperly handle the private key of the user. For this reason, this case is considered less secure and clearly flagged so accordingly in the Web application. However, from a usability point of view, it may be an acceptable compromise for some users, who are always given the choice to select which method to use anyway. Once the system has generated a keypair, the user is instructed to download its private key and delete it from the Web application.

In both cases, the user's public key is put in the appropriate location in the target VM using `libguestfs` [11]; upon VM creation, the user is then given the name of the created resource and may then use a normal passwordless `ssh` connection to login into it.

If a user subsequently chooses to create additional resources, he is given the choice to upload a new public key, to let the system to generate a new keypair, or to keep a previously used keypair.

4.1.3 Creation of the virtualized resources

Once the previous steps have been completed, Step 4, the final step of the Web application, allows the user to actually create virtual resources. When the user presses the "Create" button, as shown in Fig. 6, the WNoDeS OCCI interface will be contacted to create a single VM, or a pool of machines with the same hardware characteristics. In the latter case, the IP addresses of the created VMs will obviously be different. Once the VMs are ready, the user will be notified of their IP addresses.



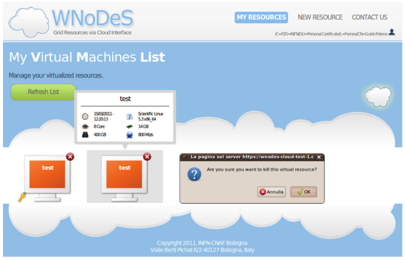
Current VO: cms (Change VO)

Give a name to your virtual machine, (max 13 characters): isgc

How many virtual machines you want to create?: 5

NOTE: If you choose to create more than one instance, they will all have the same hw configuration. The names will include an incremental suffix (e.g. "instancename1", "instancename2"...)

Figure 6: VM creation.



/C=IT/O=INFN/OU=Personal Certificate/L=Ferrara/CN=Guido Potena

My Virtual Machines List

Manage your virtualized resources.

test

15/03/2011 - 12:20:13 | Scientific Linux 5.3 x86_64
8 Core | 14 GB
400 GB | 800 Mb/s

La pagina sul server https://wnodes-cloud-test-1.c

Are you sure you want to kill this virtual resource?

Copyright 2011, INFN-CNAF Bologna
Viale Berti Pichat 6/2 40127 Bologna, Italy

Figure 7: VM display and kill.

In Fig. 7 two VMs are shown. The instantiated VMs will continue to exist in the WNoDeS system until explicitly destroyed, or until a maximum wallclock time previously agreed between user and resource provider is reached.

5. WNoDeS Web-based Administration

Besides the possibility described in the previous sections to instantiate VMs, the WNoDeS web portal can also be used by authorized users to manage WNoDeS VMs. Command-line based tools are available in WNoDeS for fine-grained control of the WNoDeS system. The goal of this section of the portal is to allow WNoDeS administrators to have an overall graphical representation of the running VMs, monitoring their usage. The main difficulty here is related to the potentially enormous number of VMs that may need to be visualized. In order to reduce the amount of space needed for such a representation, *treemaps* have been used.

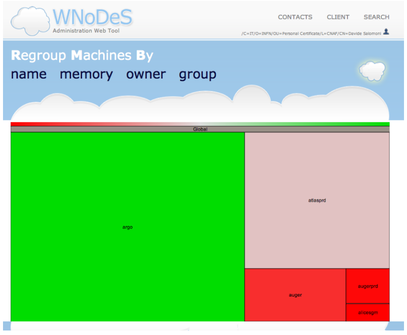
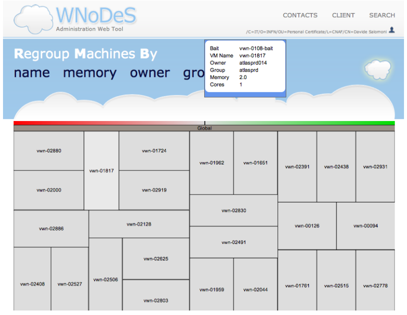


Figure 8: VMs clustered by group.



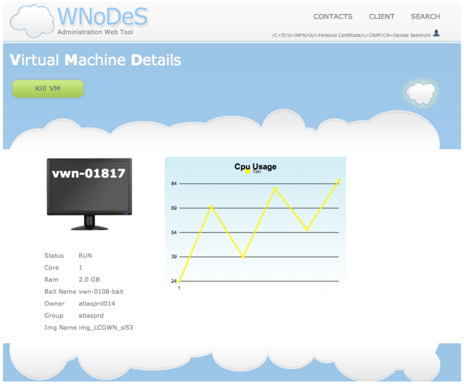
Bait	vwn-0108-bait
VM Name	vwn-01817
Owner	atlasprd014
Group	atlasprd
Memory	2.0
Cores	1

Figure 9: VMs belonging to a given group.

VMs may be grouped by name, memory used, owner or group. Fig. 8 shows a representation of running VMs clustered by group. Clicking on any of the rectangles (for example, on the *atlasprd* one) opens up a new page (Fig. 9) showing all VMs belonging to the group selected. Further details are available: clicking on any of the VMs opens up a page (Fig. 10) showing details for the VM selected, providing a simple monitoring system. It is also possible to monitor key components of the WNoDeS system like the WNoDeS Name Server, and start and stop WNoDeS services.

6. Conclusions

The WNoDeS Web portal for cloud access and for resource management is a key component of the WNoDeS framework. Through this Web application, we exploit the flexibility of WNoDeS to let users instantiate their own resources out of a common resource pool, used to seamlessly support local, Grid or Cloud requests. Although WNoDeS also provides an OCCI-compliant API to programmatically instantiate cloud resources, the Web portal offers a much simpler user experience for the self-provisioning of resources. Since it can be easily integrated with existing authentication, authorization and accounting solutions, the WNoDeS Web portal may be used by a site to offer novel added-value services to existing and new customers alike. In its current state, the WNoDeS Web portal supports users already belonging to Grid VOs. This eliminates the need for Grid users to get additional credentials to access cloud resources and - through the integration of the Argus middleware - allows sites and VOs to define fine-grained authorization policies. Future developments of the portal will include the possibility



Status	RUN
Core	1
Ram	2.0 GB
Bait Name	vwn-0108-bait
Owner	atlasprd014
Group	atlasprd
Img Name	img_LCGWN_sl53

Figure 10: Details of a given VM.

to authenticate users based on other methods, such as local accounts or federated identities (e.g. Kerberos- or Shibboleth-based authentication frameworks). For site administrators, the WNoDeS Web portal provides a graphical view of the state of the local WNoDeS installation, allowing simple monitoring and management of the WNoDeS resources.

WNoDeS is a flexible and scalable system, allowing access to resources provided by a computing center via local, Grid or Cloud interfaces. The WNoDeS Web portal, scheduled for release in the next major WNoDeS version (WNoDeS 2, expected by Fall 2011), provides on the one hand management capabilities to local site administrators beyond those provided by standard WNoDeS command-line tools; on the other hand, it gives users the possibility to rapidly self-provision their own systems. Sites using WNoDeS may use the portal to offer new services to their customers, billing them according to flexible policies. From an implementation point of view, WNoDeS and specifically its Web-based portal reuses several proven technologies, like Linux KVM, Java Spring, VOMS and Argus. Future developments may include VM image management, support for additional authentication methods and integration into a generic submission portal, thus expanding usability further and exploiting the use of existing e-Infrastructures and resources.

References

[1] WNoDeS Website: http://web.infn.it/wnodes

[2] INFN-CNAF Website: http://www.cnaf.infn.it

[3] LHC Website: http://lhc.web.cern.ch/lhc

[4] KVM Website: http://www.linux-kvm.org

[5] VOMS: Virtual Organization Membership Service, http://www.globus.org/grid_software/security/voms.php

[6] Argus Authoriztion Service Website: https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework

[7] The Open Cloud Computing Interface: http://occi-wg.org/

[8] The Spring.NET framework: http://www.springframework.net/

[9] Hibernate Website: http://www.hibernate.org/

[10] FreeMarker website: http://freemarker.sourceforge.net/

[11] libguestfs Website: http://libguestfs.org/

[12] Kernel Samepage Merging: http://www.linux-kvm.org/page/KSM