OpenNebula.org

OpenNebula 4.6 Release Notes

Release 4.6

OpenNebula Project

May 09, 2014

RELEASE NOTES 4.6

1.1 What's New

OpenNebula 4.6 introduces important improvements in several areas. The provisioning model has been greatly simplify by supplementing user groups with resource providers. This extended model, the Virtual Data Center, offers an integrated and comprehensive framework for resource allocation and isolation.

Another important new feature has taken place in the OpenNebula core. It has undergone a minor re-design of its internal data model to allow federation of OpenNebula daemons. With OpenNebula Carina your users can access resource providers from multiple data-centers in a federated way.

With Carina the OpenNebula team has started a journey to deliver a more intuitive and simpler provisioning experience for users. Our goal is level the final user usability with the system administration and operation ones. OpenNebula Carina brings a new cloud provisioning portal with a simpler, clean and intuitive design. The Sunstone graphical interface has also been tweaked to help implement common cloud admin workflows and improved in order to support the new Marketplace version, which makes even easier for a user to get a virtual application up and running.

Finally, some other areas has received the attention of the OpenNebula developers, like for example a better *Gluster <gluster_ds>* support through libgfapi, improved access to large pools pagination, or optionally limit the resources exposed by a host, among many others are included in Carina.

As usual OpenNebula releases are named after a Nebula. The Carina Nebula (NGC 3372) is one of the largest nebulae the sky. It can only be seen from the southern hemisphere, in the Carina constellation.

In the following list you can check the highlights of OpenNebula 4.6 Carina organised by component (a detailed list of changes can be found here):

1.1.1 OpenNebula Core: Virtual Data Centers

The provisioning model of OpenNebula 4.6 has been simplified to provide an integrated and comprehensive framework for resource allocation and isolation in federated data centers and hybrid cloud deployments:

- **User Groups** can be assigned one or more **resource providers**. Resource providers are defined as a cluster of servers, virtual networs, datastores and public clouds for cloud bursting in an OpenNebula zone. Read more in the *Users and Groups Management Guide*.

- A special **administration group** can be defined to manage specific aspects of the group like user management or appliances definition. Read more in the *Managing Users and Groups* guide.

- **Sunstone views** for new groups can be dynamically defined without the need of modifying the Sunstone configuration files. More information in the *Sunstone Views* guide.

- Groups can now be tagged with custom attributes. Read more in the *Managing Users and Groups* guide.

1.1.2 OpenNebula Core: Federation & Zones

Federation has been integrated in OpenNebula core:

- Users can seamlessly provision virtual machines from multiple zones with an integrated interface both in Sunstone and CLI.

- A **new tool set** has been developed to upgrade, integrate new zones and import existing zones into an OpenNebula federation. Read more in the *Federation Configuration* guide.

- **Integrated zone management** in OpenNebula core. Read more about this in the *Data Center Federation* guide.

- **Redesigned data model** to minimize replication data across zones and to tolerate large latencies. Read more about this in the *Data Center Federation* guide.

1.1.3 OpenNebula Core: Usability & Performance Enhancements

There has been several improvements for end-users:

- **Datastore management** improved with the ability to redefine some configuration attributes and default datastores. Take a look at the *Filesystem Datastore* guide to see the list of attributes.

- **Network management** also allows to change configuration attributes by updating the template. This includes `VLAN_ID`, `BRIDGE`, `VLAN` and `PHYDEV`. See the *Managing Virtual Networks* guide for more information. Network leases can be also deleted on hold.

- **Optionally limit resources** exposed by host or a cluster, see *Scheduler* for more information.

- **Support for pool pagination**, which makes OpenNebula interfaces such as the CLI and Sunstone more responsive and improves scalability. See the *Scalability* guide for more information.

1.1.4 OpenNebula Drivers

The back-end of OpenNebula has been also improved in several areas, as described below:

Storage Drivers

- **Improved Support for Ceph**, includeing RBD format 2 and direct support for CDROM devices, more info *here*.

- **Gluster support using libgfapi**, more info *here*.

Virtualization Drivers

- **KVM Hypervisor improvements** like tweakable SPICE parameters, KVM Hyper-V Enhancements, `LOCALTIME` and `MACHINE` parameters support , see the *Template* guide for more information.

- **Support for Xen FEATURES parameters**, which includes options like `PAE`, `ACPI`, `APIC`, etc. See the *Template* guide for more information.

1.1.5 Sunstone: New Cloud View

- New **cloud view with a simpler, clean and intuitive design**

1.1.6 Sunstone: Usability Enhancements

- **Updated UI Library** providing a new look.

- The **Sunstone View** for each user can be **specified in the User template or in the new Group template**, more info in the *Sunstone Views* guide.

- **Support for Zone selection**, which allows users to switch between OpenNebula Zones from within the same Sunstone.

- General usability enhancements for all the resources.

1.1.7 AppMarket

- **Native support of the new Marketplace version** by the CLI and Sunstone, which provides the possibility of importing multi-disk virtual machines. Read more in the https://github.com/OpenNebula/addon-appmarket.

- Support for **importing OVAs** processed by the **AppMarket Worker**. Read more here.

1.2 Features

This section describes the **detailed features and functionality of OpenNebula** for the management of private clouds and datacenter virtualization(*). It includes links to the different parts of the documentation and the web site that provide extended information about each feature. We also provide a summarized table of key features.

1.2.1 Powerful User Security Management

- Secure and efficient *Users and Groups Subsystem* for authentication and authorization of requests with complete functionality for user management: create, delete, show...

- *Pluggable authentication and authorization* based on *passwords*, *ssh rsa keypairs*, *X509 certificates*, *LDAP* or *Active Directory*

- Special authentication mechanisms for *SunStone (OpenNebula GUI)* and the *Cloud Services (EC2 and OCCI)*

- Authorization framework with *fine-grained ACLs* that allows multiple-role support for different types of users and administrators, delegated control to authorized users, secure isolated multi-tenant environments, and easy resource (VM template, VM image, VM instance, virtual network and host) sharing

1.2.2 Advanced Multi-tenancy with Group Management

- Administrators can *groups users* into organizations that can represent different projects, division...

- Each group have *configurable access to shared resources* so enabling a multi-tenant environment with multiple groups sharing the same infrastructure

- Configuration of special *users that are restricted to public cloud APIs* (e.g. EC2 or OCCI)

- Complete functionality for management of groups: create, delete, show...

- Multiple group support, with the ability to define primary and secondary groups.

1.2.3 On-demand Provision of Virtual Data Centers

- A vDC is a fully-isolated virtual infrastructure environment where a Group of users, optionally under the control of the vDC admin, can create and manage compute and storage capacity.

- User Groups can be assigned one or more resource providers. Resource providers are defined as a cluster of servers, virtual networks, datastores and public clouds for cloud bursting in an OpenNebula zone. Read more in the *Users and Groups Management Guide*.

- A special administration group can be defined to manage specific aspects of the group like user management or appliances definition. Read more in the *Managing Users and Groups* guide.

- Sunstone views for new groups can be dynamically defined without the need of modifying the Sunstone configuration files. More information in the *Sunstone Views* guide.

- Groups can now be tagged with custom attributes. Read more in the *Managing Users and Groups* guide.

1.2.4 Advanced Control and Monitoring of Virtual Infrastructure

- *Image Repository Subsystem* with catalog and complete functionality for VM image management: list, publish, unpublish, show, enable, disable, register, update, saveas, delete, clone...

- *Template Repository Subsystem* with catalog and complete functionality for VM template management: add, delete, list, duplicate...

- *Full control of VM instance life-cycle* and complete functionality for VM instance management: submit, deploy, migrate, livemigrate, reschedule, stop, save, resume, cancel, shutdown, restart, reboot, delete, monitor, list, power-on, power-off,...

- Advanced functionality for VM dynamic management like *system and disk snapshotting*, *capacity resizing*, or *NIC hotplugging*

- *Programmable VM operations*, so allowing users to schedule actions

- Volume hotplugging to easily hot plug a volatile disk created on-the-fly or an existing image from a Datastore to a running VM

- *Broad network virtualization capabilities* with traffic isolation, ranged or fixed networks, definition of generic attributes to define multi-tier services consisting of groups of inter-connected VMs, and complete functionality for virtual network management to interconnect VM instances: create, delete, monitor, list...

- *IPv6 support* with definition site and global unicast addresses

- Configurable *system accounting statistics* to visualize and report resource usage data, to allow their integration with chargeback and billing platforms, or to guarantee fair share of resources among users

- Tagging of users, VM images and virtual networks with arbitrary metadata that can be later used by other components

- *User defined VM tags* to simplify VM management and to store application specific data

- *Plain files datastore* to store kernels, ramdisks and files to be used in context. The whole set of OpenNebula features applies, e.g. ACLs, ownership...

1.2.5 Complete Virtual Machine Configuration

- Complete *definition of VM attributes and requirements*

- Support for automatic configuration of VMs with advanced *contextualization mechanisms*

- *Cloud-init* support

- *Hook Manager* to trigger administration scripts upon VM state change

- Wide range of guest operating system including Microsoft Windows and Linux

- *Flexible network defintion*

- *Configuration of firewall for VMs* to specify a set of black/white TCP/UDP ports

1.2.6 Advanced Control and Monitoring of Physical Infrastructure

- *Configurable to deploy public, private and hybrid clouds*

- *Host Management Subsystem* with complete functionality for management of physical hosts: create, delete, enable, disable, monitor, list...

- Dynamic creation of *clusters* as a logical set of physical resources, namely: hosts, networks and data stores, within each zone

- Highly scalable and extensible built-in *monitoring subsystem*

1.2.7 Broad Commodity and Enterprise Platform Support

- Hypervisor agnostic *Virtualization Subsystem* with broad hypervisor support (*Xen*, *KVM* and *VMware*), centralized management of environments with multiple hypervisors, and support for multiple hypervisors within the same physical box

- *Storage Subsystem* with support for multiple data stores to balance I/O operations between storage servers, or to define different SLA policies (e.g. backup) and performance features for different VM types or users

- *Storage Subsystem* supporting any backend configuration with different datastore types: *file system datastore*, to store disk images in a file form and with image transferring using ssh or shared file systems (NFS, GlusterFS, Lustre...), *LVM* to store disk images in a block device form, *Ceph* for distributed block device, and *VMware datastore* specialized for the VMware hypervisor that handle the vmdk format and with support for VMFS

- Flexible *Network Subsystem* with integration with *Ebtable*, *Open vSwitch* and *802.1Q tagging*

- *Virtual Router* fully integrated with OpenNebula to provide basic L3 services like NATting, DHCP, DNS...

1.2.8 Distributed Resource Optimization

- Powerful and flexible *requirement/rank matchmaker scheduler* providing automatic initial VM placement for the definition of workload and resource-aware allocation policies such as packing, striping, load-aware, affinity-aware...

- *Advanced requirement expressions* with cluster attributes for VM placement, affinity policies, any host attribute for scheduling expressions, and scheduler feedback through VM tags

- Powerful and flexible *requirement/rank matchmaker scheduler* for storage load balancing to distribute efficiently the I/O of the VMs across different disks, LUNs or several storage backends

- *Resource quota management* to allocate, track and limit computing, storage and networking resource utilization

- Support for *cgroups* on KVM to enforce VM CPU usage as described in the VM Template

1.2.9 Centralized Management of Multiple Zones

- Federation of multiple OpenNebula zones for scalability, isolation or multiple-site support

- Users can seamlessly provision virtual machines from multiple zones with an integrated interface both in Sunstone and CLI.

- A new tool set has been developed to upgrade, integrate new zones and import existing zones into an OpenNebula federation. Read more in the *Federation Configuration* guide.

- Integrated zone management in OpenNebula core. Read more about this in the *Data Center Federation* guide.

- Redesigned data model to minimize replication data across zones and to tolerate large latencies. Read more about this in the *Data Center Federation* guide.

- Complete functionality for management of zones: create, delete, show, list...

1.2.10 High Availability

- Persistent database backend with support for high availability configurations

- *Configurable behavior in the event of host, VM, or OpenNebula instance failure to provide an easy to use and cost-effective failover solution*

- Support for *high availability architectures*

1.2.11 Community Virtual Appliance Marketplace

- Marketplace with an online catalog where individuals and organizations can quickly distribute and deploy virtual appliances ready-to-run on OpenNebula cloud environments

- *Marketplace is fully integrated with OpenNebula* so any user of an OpenNebula cloud can find and deploy virtual appliances in a single click through familiar tools like the SunStone GUI or the OpenNebula CLI

- Support for importing OVAs processed by the AppMarket Worker. Read more here.

1.2.12 Management of Multi-tier Applications

- *Automatic execution of multi-tiered applications* with complete functionality for the management of groups of virtual machines as a single entity: list, delete, scale up, scale down, shutdown... and the management of Service Templates: create, show, delete, instantiate...

- *Automatic deployment and undeployment of Virtual Machines* according to their dependencies in the Service Template

- Provide configurable services from a catalog and self-service portal

- Enable tight, efficient administrative control

- Complete integration with the OpenNebula's User Security Management system

- Computing resources can be tracked and limited using OpenNebula's *Resource Quota Management*

- *Automatic scaling of multi-tiered applications* according to performance metrics and time schedule

1.2.13 Gain Insight into Cloud Applications

- *OneGate allows Virtual Machine guests to push monitoring information to OpenNebula*

- With a security token the VMs can call back home and report guest and/or application status in a simple way, that can be easily queried through OpenNebula interfaces (Sunstone, CLI or API).

- Users and administrators can use it to gather metrics, detect problems in their applications, and trigger *OneFlow auto-scaling rules*

1.2.14 Hybrid Cloud Computing and Cloud Bursting

- *Extension of the local private infrastructure with resources from remote clouds*

- *Support for Amazon EC2* with most of the EC2 features like tags, security groups or VPC; and simultaneous access to multiple remote clouds

1.2.15 Standard Cloud Interfaces and Simple Provisioning Portal for Cloud Consumers

- *Transform your local infrastructure into a public cloud by exposing REST-based interfaces*

- *OGF OCCI service*, the emerging cloud API standard, and *client tools*

- *AWS EC2 API service*, the de facto cloud API standard, with *compatibility with EC2 ecosystem tools* and *client tools*

- Support for simultaneously exposing multiple cloud APIs

- *Provisioning portal implemented as a user view of Sunstone* to allow non-IT end users to easily create, deploy and manage compute, storage and network resources

1.2.16 Rich Command Line and Web Interfaces for Cloud Administrators

- *Unix-like Command Line Interface* to manage all resources: users, VM images, VM templates, VM instances, virtual networks, zones, VDCs, physical hosts, accounting, authentication, authorization...

- *Easy-to-use Sunstone Graphical Interface* providing usage graphics and statistics with cloudwatch-like functionality, VNC support, different system views for different roles, catalog access, multiple-zone management...

- *Sunstone is easily customizable* to define multiple cloud views for different user groups

1.2.17 Multiple Deployment Options

- *Easy to install and update* with packages for most common Linux distributions

- Available in most popular Linux distributions

- *Optional building from source code*

- *System features a small footprint*, less than 10Mb

- *Detailed log files* with *syslog support* for the different components that maintain a record of significant changes

1.2.18 Easy Extension and Integration

- Modular and extensible architecture to fit into any existing datacenter

- Customizable drivers for the main subsystems to easily leverage existing IT infrastructure and system management products: *Virtualization*, *Storage*, *Monitoring*, *Network*, *Auth* and *Hybrid Cloud*

- New drivers can be easily written in any language

- Plugin support to easily extend SunStone Graphical Interface with additional tabs to better integrate Cloud and VM management with each site own operations and tools

- Easily customizable self-service portal for cloud consumers

- *Configuration and tuning parameters* to adjust behavior of the cloud management instance to the requirements of the environment and use cases

- Fully open-source technology available under Apache license

- Powerful and extensible low-level cloud API in *Ruby* and *JAVA* and *XMLRPC API*

- OpenNebula Add-on Catalog with components enhancing the functionality provided by OpenNebula

1.2.19 Reliability, Efficiency and Massive Scalability

- Automated testing process for functionality, scalability, performance, robustness and stability

- Technology matured through an active and engaged community

- Proven on large scale infrastructures consisting of tens of thousands of cores and VMs

- Highly scalable database back-end with support for *MySQL* and SQLite

- Virtualization drivers adjusted for maximum scalability

- Very efficient core developed in C++ language

(*) *Because OpenNebula leverages the functionality exposed by the underlying platform services, its functionality and performance may be affected by the limitations imposed by those services.*

- *The list of features may change on the different platform configurations*

- *Not all platform configurations exhibit a similar performance and stability*

- *The features may change to offer users more features and integration with other virtualization and cloud components*

- *The features may change due to changes in the functionality provided by underlying virtualization services*

1.3 Platform Notes

This page will show you the specific considerations at the time of using an OpenNebula cloud, according to the different supported platforms.

This is the list of the individual platform components that have been through the complete OpenNebula Quality Assurance and Certification Process.

Certified Platform Component	Version
RedHat Enterprise Linux	6.5
Ubuntu Server	12.04 (LTS) & 14.04 (LTS)
SUSE Linux Enterprise	12.3
CentOS	6.5
openSUSE	12.3
Debian	7.1
VMware	ESX 5.0 & ESX 5.1
XEN	3.2 & 4.2
KVM	Supported version that is included in the kernel for the Linux distribution

1.3.1 All Front-Ends

- xmlrpc tuning parameters (MAX_CONN, MAX_CONN_BACKLOG, KEEPALIVE_TIMEOUT, KEEPALIVE_MAX_CONN and TIMEOUT) are only available with packages distributed by us as they are compiled with a newer xmlrpc-c library.

- for **cloud bursting**, a newer nokogiri gem than the on packed by current distros is required. If you are planning to use cloud bursting, you need to install nokogiri >= 1.4.4 prior to run `install_gems`

```
# sudo gem install nokogiri -v 1.4.4
```

ESX 5.x as VMware Node

- to accomplish disk hotplugging and nic hotplugging (ignore the first bullet for the latter)

    - disks need to be attached through SCSI, so their images should have a DEV_PREFIX="sd"

    - VM template that will permit SCSI disk attaches afterwards needs to have an explicitly defined SCSI controller:

```
RAW=[TYPE = "vmware",
    DATA = "<devices><controller type='scsi' index='0' model='lsilogic'/></devices>"]
```

- to use SCSI disk based VMs, it is usually a good idea to explicitly declare the PCI bridges. This can be accomplished with the following added to the VM template:

```
FEATURES=[PCIBRIDGE="1"]
```

- to accomplish hot migration (through vMotion)

    - VM needs to have all network card model with model "E1000"

CentOS 6.5 as KVM Node

- to accomplish disk hotplugging:

    - disks need to be attached through SCSI, so their images should have a DEV_PREFIX="sd"

    - VM template that will permit SCSI disk attaches afterwards needs to have an explicitly defined SCSI controller:

```
RAW=[TYPE = "kvm",
    DATA = "<devices><controller type='scsi' index='0' model='virtio-scsi'></controller></devices>"]
```

- due to libvirt version < = 0.10.2, there is a bug in libvrit/qemu attac/detach nic functionality that prevents the reuse of net IDs. This means that after a successful attach/detach NIC, a new attach will fail.

Ubuntu 12.04 as KVM Node

- due to libvirt version < = 0.10.2, there is a bug in libvrit/qemu attac/detach nic functionality that prevents the reuse of net IDs. This means that after a successful attach/detach NIC, a new attach will fail.

1.3.2 CentOS 6.5 Usage Platform Notes

Because home directory of oneadmin is located in `/var`, it violates SELinux default policy. So in ssh passwordless configuration you should disable SELinux by setting `SELINUX=disabled` in `/etc/selinux/config`.

1.3.3 Debian Platform Notes

Debian Lenny as Xen 3 Node

- The xen packages on Debian Lenny seem to be broken, and they don't work with the tap:aio interface. A workaround for this problem is the following:

```
# ln -s /usr/lib/xen-3.2-1/bin/tapdisk /usr/sbin
# echo xenblktap >> /etc/modules
# reboot
```

1.3.4 openSUSE 12.3 Platform Notes

- Limited startup scripts → only for OpenNebula and Sunstone services

1.3.5 Ubuntu 12.04 Platform Notes

- Limited startup scripts → only for OpenNebula service
- Ubuntu12.04 presents libvirt 0.9.8. We recommend updating (manually, there are no packages) to 0.10.2 to use the AttachNic and DetachNic functionality.

1.3.6 Ubuntu 14.04 Platform Notes

- Limited startup scripts → only for OpenNebula service

1.3.7 Unsupported Platforms Notes

Installing on ArchLinux

OpenNebula is available at the Arch User Repository (AUR), please check the opennebula package page.

Installing on Gentoo

You need to compile the xmlrpc-c package with threads support, as:

```
USE="threads" emerge xmlrpc-c
```

1.4 Compatibility Guide

This guide is aimed at OpenNebula 4.4 users and administrators who want to upgrade to the latest version. The following sections summarize the new features and usage changes that should be taken into account, or prone to cause confusion. You can check the upgrade process in the following *guide*

Visit the *Features list* and the Release Notes for a comprehensive list of what's new in OpenNebula 4.6.

1.4.1 OpenNebula Administrators and Users

oZones and Federation

oZones

The **oZones** component has been deprecated, its funcitonality has been integrated into the main OpenNebula daemon. You can read more about the new **OpenNebula Federation** capabilities *here*. In a Federation, different OpenNebula instances share the same user accounts, groups, and permissions configuration.

Virtual Data Centers (vDC)

The **onevdc** command was part of the now deprecated **oZones** component. In 4.4, vDC provided a way to assign physical resources to groups, and to define a group administrator. These features have been extended and integrated into the core Group, so everything is managed through Sunstone and the CLI. You can read more about Groups, vDC, and Resource Providers in the *Group Management guides* and the *Quickstart guide*.

Resource Providers

In previous versions, administrators could decide the physical resources exposed to each group using *ACL rules*. In 4.6, this process has been simplified merging the oZones vDC functionality in the core.

There is a new concept: Resource Provider. A Resource Provider is an OpenNebula *cluster* (set of physical hosts and associated datastores and virtual networks) from a particular Zone (an OpenNebula instance). You can *assign/remove Resource Providers* to each group, and internally that will create the ACL rules that you had to manually manage in 4.4.

When you upgrade from a previous version, the groups will not have any Resource Provider assigned. The existing ACL Rules are preserved, but they will not be interpreted as Resource Providers.

ACL Rules in a Federation

Each *ACL Rule* now can define the Zone(s) where it applies.

```
$ oneacl create "@103 IMAGE/@100 USE #0"
ID: 4

$ oneacl create "@103 IMAGE/@102 USE *"
ID: 5

$ oneacl list
   ID     USER RES_VHNIUTGDCOZ    RID OPE_UMAC   ZONE
    4     @103    ---I-------   @100     u---     #0
    5     @103    ---I-------   @102     u---      *
```

Groups

Group resources now have a template that can be edited, via Sunstone or with the `onegroup update` command.

The 4.4 configuration file `group.default` defined the ACL rules to create when new groups were created. This was the default `group.default` file:

```
# This rule allows users in the new group to create common resources
VM+NET+IMAGE+TEMPLATE/* CREATE
# This rule allows users in the group to deploy VMs in any host in the cloud
HOST/* MANAGE
```

This file does not exist in 4.6, and the ACL rules are controlled with the *–resources option* and Resource Providers.

> `-r, --resources:` Defines the resources that can be created by group users (VM+IMAGE+TEMPLATE by default)

Users in new groups cannot deploy VMs until the admin assigns a *Resource Provider*. To replicate the default 4.4 behaviour, assign the resource provider ALL:

```
$ onegroup add_provider <group_id> 0 ALL
```

Note: Note the difference in the default resources: users cannot create VNETs by default anymore.

Sunstone

The Sunstone web interface has undergone a major redesign to make it more intuitive to use. There is more information available, new Views, and better creations wizards. You will not experience any incompatibility interaction with previous resources.

Sunstone Cloud View

- The Self-service Cloud View in OpenNebula 4.4 was a trimmed down version of the normal Sunstone. Now the users access a newly designed *Cloud View*.

- The Templates for the 4.4 self-service cloud view **are not compatible** with the new 4.6 Cloud View.

  - In 4.4, self-service Templates were defined without the Disk, that the user could select at launch time.

  - For 4.6, Templates are complete i.e. can be instantiated without adding any attributes.

    The rationale behind this is:

  - The same Templates work for both the Cloud View and for administrators/advanced user views. There is no need to create specific Templates.

  - Allowing users to combine Templates with Images may end in non-functional VMs. For example, if you setup OS/ARCH to be 64bits a 64bit Image is needed.

  - It is quite difficult to combine the self-service Templates in 4.4 with advanced features like hybrid clouds. You need to tie somehow the local Image with the remote AMI. That can only be done at the Template level.

- The Cloud View Users can now save the changes they make to their VMs. This is a new action that combines a deferred disk snapshot with a template clone. Read more in the *vDC Admin View guide*.

Sunstone Configuration

- Available sunstone views for each user are now defined group-wise, instead of specifying it in `sunstone-views.yaml` (this file keeps the default view). Note, that there is no need to restart the Sunstone server when views are updated.

- The new Cloud view features predefined Instance types, that specify capacity values to instantiate the predefined VM templates. The instance types can be defined in `sunstone-server.conf`.

- The `admin.yaml` and `user.yaml` views now have OneFlow visible by default. If you don't have OneFlow running, you'll see the message: 'Cannot connect to OneFlow server'. *Follow this section* to know more. Flow can be disables in `sunstone-server.conf`.

- The communication between clients and Sunstone server has been reduced in this release to increase the overall performance (server-side). The active tab panel periodically refresh its contents. This behavior is controlled by the `autorefresh` option in the yaml view definition file.

- Views should be compatible with Sunstone 4.6, note that `autorefresh` will be set to `false` for existing views.

Storage

- The Datastore *BASE_PATH can be edited* after the Datastore creation. You can also modify the base path of the default Datastores (0,1,2).

- Default Datastores (0,1,2) could not be deleted, or assigned to a Cluster up to 4.4. This limitation has been removed for the 4.6 release.

- OpenNebula can now operate with RBD Format 1 and RBD Format 2. RBD Format 2 brings many advantages, like creating new clones based on snapshots which runs a lot faster. If you want to take advantage of this you will need to manually convert your previously created images from RBD Format 1 to Format 2.

- For Ceph Datastores, you can specify the extra args sent to `qemu-img convert` in the `/var/lib/one/remotes/datastore/ceph/ceph.conf'''file.  Adding ''-O rbd` is recommended depening on the RBD version.

- KVM machines support `libgfapi` to connect to disk images greatly improving the IO performance over FUSE mounts. The information about is is at the *GlusterFS Datastore guide*

Monitoring

There is a new probe that kills wild `collectd-client` processes in case more than one is running.

Scheduling

Deprecated attribute in sched.conf, `HYPERVISOR_MEM`.

> `HYPERVISOR_MEM`: Fraction of total MEMORY reserved for the hypervisor. E.g. 0.1 means that only 90% of the total MEMORY will be used.

The admin can now *update the Host information* to set a limit on the CPU and MEMORY available to OpenNebula. See the *Scheduler Guide* for more information.

This functionality is somewhat similar the 4.4 `HYPERVISOR_MEM` attribute in `sched.conf`. But it is more useful since the limitation applies to the complete OpenNebula system, not only to the Scheduler, and because it can be set for each Host individually.

- A new attribute has been included in sched.conf `MESSAGE_SIZE` to set the buffer size in bytes for XML-RPC responses. This can be increased in large clouds with a great number of VMs.

AppMarket

In the new Appmarket version (>= 2) all the appliances can have multiple disks. This however does not affect the Sunstone workflow.

Contextualization

- VM hostname can be specified in the ccontext section with `SET_HOSTNAME` or can be automatically set from DNS with `DNS_HOSTNAME=YES`.

- Latest versions of cloud-init can be used if the network is disabled in `USER_DATA`. For example:

```
CONTEXT=[
  NETWORK="YES",
  SSH_PUBLIC_KEY="$USER[SSH_PUBLIC_KEY]",
  USER_DATA="#cloud-config
  bootcmd:
    - ifdown -a"
]
```

Accounting: oneacct command

The `oneacct` command now accepts the following options:

- `--csv`: Writes the table in csv format

- `--describe`: Describes the list columns

- `-l, --list`: Selects the columns to display with list command

For example:

```
$ oneacct --list UID,HOSTNAME,CPU --csv
```

Virtual Networks

- The following Virtual Network attributes can now be updated after the VNet creation:

  - `PHYDEV`

  - `VLAN`

  - `VLAN_ID`

  - `BRIDGE`

- Leases on hold can now be deleted.

- Up to 4.4, a Virtual Network could be deleted at any time even if there were VMs using IPs from that network. Now a Virtual Network cannot be deleted if there are leases in use.

oned.conf

New attributes in *oned.conf*:

- `VM_INDIVIDUAL_MONITORING`: VM monitoring information is obtained along with the host information. For some custom monitor drivers you may need activate the individual VM monitoring process.

- `FEDERATION`: Attributes to control the *federation config*.

- `MESSAGE_SIZE`: Buffer size in bytes for XML-RPC responses. Only relevant for federation slave zones.

- `RPC_LOG`: Create a separated log file for xml-rpc requests, in /var/log/one/one_xmlrpc.log.

- `DEFAULT_CDROM_DEVICE_PREFIX`: Same as `DEFAULT_DEVICE_PREFIX` but for CDROM devices. Default value for DEV_PREFIX field when it is omitted in a template.

oneflow-server.conf

There is a new configuration attribute to customize the name given to the VMs created by oneflow. Read the *OneFlow Server Configuration guide* for more information

`:vm_name_template`: Default name for the Virtual Machines created by oneflow. You can use any of the following placeholders

- $SERVICE_ID

- $SERVICE_NAME

- $ROLE_NAME

- $VM_NUMBER

KVM

There are new parameters for KVM machines. They can be seen in more detail at the *VM Template reference guide*:

- `hyperv` feature to give better support to Windows machines. Its paramaters can be changed in the driver configuration file (`/etc/one/vmm_exec/vmm_exec_kvm.conf`).

- `localtime` feature so the clock reported to the VM is in local time instead of UTC.

- When selecting spice support some other parameters are added to the VM. These can be changed in the driver configuration file (`/etc/one/vmm_exec/vmm_exec_kvm.conf`).

- Add default `GRAPHICS` section parameters, configurable in the driver configuration file (`/etc/one/vmm_exec/vmm_exec_kvm.conf`).

- `machine` option in the `OS` section. This is useful for migration between different host OS versions and to select other chipset than the default one.

Xen

There are new parameters for Xen machines. They can be seen in more detail at the *VM Template reference guide*:

- `localtime` feature so the clock reported to the VM is in local time instead of UTC.

- New VM features `pae`, `acpi` and `apic` for Xen HVM.

- Device model can now be set as a VM feature.

- The pointer device can be a tablet usb. This makes VNC mouse handling more precise.

1.4.2 Developers and Integrators

Monitoring

- Individual VM monitoring has been disabled by default for stock monitoring drivers. These drivers include VM information along with the hypervisor one. As the VM information is also obtained (in general) through the hypervisor, a failure may collapse VM actions. If you have develop a driver that relies on individual VM monitoring, you can enable it in `oned.conf` (attribute `VM_INDIVIDUAL_MONITORING`)

Ruby OCA

- `OpenNebula::Client` has a new option `http_proxy` to specify the proxy.

- `OpenNebula::*Pool.get_hash` can be used to retrieve a hash with the pool. For Pools that support it the query to `oned` can be paginated.

- OCA will use the gem Ox when it is installed. This makes parsing both XML-RPC and xml responses much faster.

XML-RPC API

- `Pool.info` calls with `end_id` < -1 can be used to paginate responses. The offset is specified as `start_id` and the number of objects * -1 as `end_id`. For example to get 15 VMs skiping 250 the call will be:

```
vm_pool.info(-1, 250, -15)
```

- New api calls:
    - `one.group.update`: Replaces the group template contents.
    - `one.group.addprovider`: Adds a resource provider to the group.
    - `one.group.delprovider`: Deletes a resource provider from the group.
    - `one.zone.allocate`: Allocates a new zone in OpenNebula.
    - `one.zone.delete`: Deletes the given zone from the pool.
    - `one.zone.update`: Replaces the zone template contents.
    - `one.zone.rename`: Renames a zone.
    - `one.zone.info`: Retrieves information for the zone.
    - `one.zonepool.info`: Retrieves information for all the zones in the pool.
- Changed api calls:
    - `one.vm.savedisk`: New optional parameter. Boolean, True to clone clone also the VM originating Template, and replace the disk with the saved image.

1.5 Known Issues

1.5.1 Core & System

- #2831 Monitoring data is added to the VM even when is not active

1.5.2 Packaging

- #2837 Sunstone start scripts may leave a running process without writting down the pid

1.5.3 Sunstone

- #2814 If a role name contains < or >, the extended role view fails to open
- #2813 Sporadic Uncaught TypeError in vm list callback
- #2804 Wizard multple selection dataTables do not highlight element outside the current page
- #2799 Template update wizard: intro calls create, not update

1.6 Acknowledgements

The OpenNebula project would like to thank the community members and users who have contributed to this software release by being active with the discussions, answering user questions, or providing patches for bugfixes, features and documentation.

The new features for VDCs, Federations and OVA support in the Marketplace introduced in OpenNebula 4.6 were funded by Produban in the context of the Fund a Feature Program.

1.7 Upgrading from OpenNebula 4.4.x

This guide describes the installation procedure for systems that are already running a 4.4.x OpenNebula. The upgrade will preserve all current users, hosts, resources and configurations; for both Sqlite and MySQL backends.

Read the *Compatibility Guide* and Release Notes to know what is new in OpenNebula 4.6.

1.7.1 Preparation

Before proceeding, make sure you don't have any VMs in a transient state (prolog, migr, epil, save). Wait until these VMs get to a final state (runn, suspended, stopped, done). Check the *Managing Virtual Machines guide* for more information on the VM life-cycle.

Stop OpenNebula and any other related services you may have running: EC2, OCCI, and Sunstone. As `oneadmin`, in the front-end:

```
$ sunstone-server stop
$ oneflow-server stop
$ econe-server stop
$ occi-server stop
$ one stop
```

1.7.2 Backup

Backup the configuration files located in **/etc/one**. You don't need to do a manual backup of your database, the onedb command will perform one automatically.

1.7.3 Installation

Follow the *Platform Notes* and the *Installation guide*, taking into account that you will already have configured the passwordless ssh access for oneadmin.

It is highly recommended **not to keep** your current `oned.conf`, and update the `oned.conf` file shipped with Open-Nebula 4.6 to your setup. If for any reason you plan to preserve your current `oned.conf` file, read the *Compatibility Guide* and the complete oned.conf reference for *4.4* and *4.6* versions.

1.7.4 Database Upgrade

The database schema and contents are incompatible between versions. The OpenNebula daemon checks the existing DB version, and will fail to start if the version found is not the one expected, with the message 'Database version mismatch'.

You can upgrade the existing DB with the 'onedb' command. You can specify any Sqlite or MySQL database. Check the *onedb reference* for more information.

> **Warning:** Make sure at this point that OpenNebula is not running. If you installed from packages, the service may have been started automatically.

After you install the latest OpenNebula, and fix any possible conflicts in oned.conf, you can issue the 'onedb upgrade -v' command. The connection parameters have to be supplied with the command line options, see the *onedb manpage* for more information. Some examples:

```
$ onedb upgrade -v --sqlite /var/lib/one/one.db
```

```
$ onedb upgrade -v -S localhost -u oneadmin -p oneadmin -d opennebula
```

If everything goes well, you should get an output similar to this one:

```
$ onedb upgrade -v -u oneadmin -d opennebula
MySQL Password:
Version read:
Shared tables 4.4.0 : OpenNebula 4.4.0 daemon bootstrap
Local tables  4.4.0 : OpenNebula 4.4.0 daemon bootstrap

>>> Running migrators for shared tables
  > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.0_to_4.4.1.rb
  > Done in 0.00s

  > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.1_to_4.5.80.rb
  > Done in 0.75s

Database migrated from 4.4.0 to 4.5.80 (OpenNebula 4.5.80) by onedb command.

>>> Running migrators for local tables
Database already uses version 4.5.80
Total time: 0.77s
```

> **Note:** Make sure you keep the backup file. If you face any issues, the onedb command can restore this backup, but it won't downgrade databases to previous versions.

1.7.5 Check DB Consistency

After the upgrade is completed, you should run the command `onedb fsck`.

First, move the 4.4 backup file created by the upgrade command to a safe place.

```
$ mv /var/lib/one/mysql_localhost_opennebula.sql /path/for/one-backups/
```

Then execute the following command:

```
$ onedb fsck -S localhost -u oneadmin -p oneadmin -d opennebula
MySQL dump stored in /var/lib/one/mysql_localhost_opennebula.sql
Use 'onedb restore' or restore the DB using the mysql command:
mysql -u user -h server -P port db_name < backup_file

Total errors found: 0
```

1.7.6 Update the Drivers

You should be able now to start OpenNebula as usual, running 'one start' as oneadmin. At this point, execute `onehost sync` to update the new drivers in the hosts.

> **Warning:** Doing `onehost sync` is important. If the monitorization drivers are not updated, the hosts will behave erratically.

1.7.7 Testing

OpenNebula will continue the monitoring and management of your previous Hosts and VMs.

As a measure of caution, look for any error messages in oned.log, and check that all drivers are loaded successfully. After that, keep an eye on oned.log while you issue the onevm, onevnet, oneimage, oneuser, onehost **list** commands. Try also using the **show** subcommand for some resources.

1.7.8 Restoring the Previous Version

If for any reason you need to restore your previous OpenNebula, follow these steps:

- With OpenNebula 4.6 still installed, restore the DB backup using 'onedb restore -f'
- Uninstall OpenNebula 4.6, and install again your previous version.
- Copy back the backup of /etc/one you did to restore your configuration.

1.7.9 Known Issues

If the MySQL database password contains specials characters, such as @ or #, the onedb command will fail to connect to it.

The workaround is to temporarily change the oneadmin's password to an ASCII string. The set password statement can be used for this:

```
$ mysql -u oneadmin -p

mysql> SET PASSWORD = PASSWORD('newpass');
```

1.8 Upgrading from OpenNebula 4.2

This guide describes the installation procedure for systems that are already running a 4.2 OpenNebula. The upgrade will preserve all current users, hosts, resources and configurations; for both Sqlite and MySQL backends.

Read the Compatibility Guide for 4.4 and *4.6*, and the Release Notes to know what is new in OpenNebula 4.6.

> **Warning:** With the new *multi-system DS* functionality, it is now required that the system DS is also part of the cluster. If you are using System DS 0 for Hosts inside a Cluster, any VM saved (stop, suspend, undeploy) **will not be able to be resumed after the upgrade process**.

> **Warning:** Two drivers available in 4.0 are now discontinued: **ganglia** and **iscsi**.
> - **iscsi** drivers have been moved out of the main OpenNebula distribution and are available (although not supported) as an addon.
> - **ganglia** drivers have been moved out of the main OpenNebula distribution and are available (although not supported) as an addon.

1.8.1 Preparation

Before proceeding, make sure you don't have any VMs in a transient state (prolog, migr, epil, save). Wait until these VMs get to a final state (runn, suspended, stopped, done). Check the *Managing Virtual Machines guide* for more information on the VM life-cycle.

Stop OpenNebula and any other related services you may have running: EC2, OCCI, and Sunstone. As `oneadmin`, in the front-end:

```
$ sunstone-server stop
$ oneflow-server stop
$ econe-server stop
$ occi-server stop
$ one stop
```

1.8.2 Backup

Backup the configuration files located in **/etc/one**. You don't need to do a manual backup of your database, the onedb command will perform one automatically.

1.8.3 Installation

Follow the *Platform Notes* and the *Installation guide*, taking into account that you will already have configured the passwordless ssh access for oneadmin.

It is highly recommended **not to keep** your current `oned.conf`, and update the `oned.conf` file shipped with Open-Nebula 4.6 to your setup. If for any reason you plan to preserve your current `oned.conf` file, read the *Compatibility Guide* and the complete oned.conf reference for 4.2 and *4.6* versions.

1.8.4 Database Upgrade

The database schema and contents are incompatible between versions. The OpenNebula daemon checks the existing DB version, and will fail to start if the version found is not the one expected, with the message 'Database version mismatch'.

You can upgrade the existing DB with the 'onedb' command. You can specify any Sqlite or MySQL database. Check the *onedb reference* for more information.

> **Warning:** Make sure at this point that OpenNebula is not running. If you installed from packages, the service may have been started automatically.

After you install the latest OpenNebula, and fix any possible conflicts in oned.conf, you can issue the 'onedb upgrade -v' command. The connection parameters have to be supplied with the command line options, see the *onedb manpage* for more information. Some examples:

```
$ onedb upgrade -v --sqlite /var/lib/one/one.db
```

```
$ onedb upgrade -v -S localhost -u oneadmin -p oneadmin -d opennebula
```

If everything goes well, you should get an output similar to this one:

```
$ onedb upgrade -v -u oneadmin -d opennebula
MySQL Password:
Version read:
Shared tables 4.4.0 : OpenNebula 4.4.0 daemon bootstrap
Local tables  4.4.0 : OpenNebula 4.4.0 daemon bootstrap

>>> Running migrators for shared tables
  > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.0_to_4.4.1.rb
  > Done in 0.00s

  > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.1_to_4.5.80.rb
  > Done in 0.75s

Database migrated from 4.4.0 to 4.5.80 (OpenNebula 4.5.80) by onedb command.

>>> Running migrators for local tables
Database already uses version 4.5.80
Total time: 0.77s
```

If you receive the message "ATTENTION: manual intervention required", read the section *Manual Intervention Required* below.

> **Note:** Make sure you keep the backup file. If you face any issues, the onedb command can restore this backup, but it won't downgrade databases to previous versions.

1.8.5 Check DB Consistency

After the upgrade is completed, you should run the command `onedb fsck`.

First, move the 4.2 backup file created by the upgrade command to a safe place.

```
$ mv /var/lib/one/mysql_localhost_opennebula.sql /path/for/one-backups/
```

Then execute the following command:

```
$ onedb fsck -S localhost -u oneadmin -p oneadmin -d opennebula
MySQL dump stored in /var/lib/one/mysql_localhost_opennebula.sql
Use 'onedb restore' or restore the DB using the mysql command:
mysql -u user -h server -P port db_name < backup_file

Total errors found: 0
```

1.8.6 Update the Drivers

You should be able now to start OpenNebula as usual, running 'one start' as oneadmin. At this point, execute `onehost sync` to update the new drivers in the hosts.

> **Warning:** Doing `onehost sync` is important. If the monitorization drivers are not updated, the hosts will behave erratically.

1.8.7 Setting new System DS

With the new *multi-system DS* functionality, it is now required that the system DS is also part of the cluster. If you are using System DS 0 for Hosts inside a Cluster, any VM saved (stop, suspend, undeploy) **will not be able to be resumed after the upgrade process**.

You will need to have at least one system DS in each cluster. If you don't already, create new system DS with the same definition as the system DS 0 (TM_MAD driver). Depending on your setup this may or may not require additional configuration on the hosts.

You may also try to recover saved VMs (stop, suspend, undeploy) following the steps described in this thread of the users mailing list.

1.8.8 Testing

OpenNebula will continue the monitoring and management of your previous Hosts and VMs.

As a measure of caution, look for any error messages in oned.log, and check that all drivers are loaded successfully. After that, keep an eye on oned.log while you issue the onevm, onevnet, oneimage, oneuser, onehost **list** commands. Try also using the **show** subcommand for some resources.

1.8.9 Restoring the Previous Version

If for any reason you need to restore your previous OpenNebula, follow these steps:

- With OpenNebula 4.6 still installed, restore the DB backup using 'onedb restore -f'
- Uninstall OpenNebula 4.6, and install again your previous version.
- Copy back the backup of /etc/one you did to restore your configuration.

1.8.10 Known Issues

If the MySQL database password contains specials characters, such as @ or #, the onedb command will fail to connect to it.

The workaround is to temporarily change the oneadmin's password to an ASCII string. The set password statement can be used for this:

```
$ mysql -u oneadmin -p

mysql> SET PASSWORD = PASSWORD('newpass');
```

1.8.11 Manual Intervention Required

If you have a datastore configured to use a tm driver not included in the OpenNebula distribution, the onedb upgrade command will show you this message:

```
ATTENTION: manual intervention required

The Datastore <id> <name> is using the
custom TM MAD '<tm_mad>'. You will need to define new
configuration parameters in oned.conf for this driver, see
http://opennebula.org/documentation:rel4.4:upgrade
```

Since OpenNebula 4.4, each tm_mad driver has a TM_MAD_CONF section in oned.conf. If you developed the driver, it should be fairly easy to define the required information looking at the existing ones:

```
# The  configuration for each driver is defined in TM_MAD_CONF. These
# values are used when creating a new datastore and should not be modified
# since they define the datastore behaviour.
#   name      : name of the transfer driver, listed in the -d option of the
#               TM_MAD section
#   ln_target : determines how the persistent images will be cloned when
#               a new VM is instantiated.
#       NONE: The image will be linked and no more storage capacity will be used
#       SELF: The image will be cloned in the Images datastore
#       SYSTEM: The image will be cloned in the System datastore
#   clone_target : determines how the non persistent images will be
#                  cloned when a new VM is instantiated.
#       NONE: The image will be linked and no more storage capacity will be used
#       SELF: The image will be cloned in the Images datastore
#       SYSTEM: The image will be cloned in the System datastore
#   shared : determines if the storage holding the system datastore is shared
#            among the different hosts or not. Valid values: "yes" or "no"

TM_MAD_CONF = [
    name        = "lvm",
    ln_target   = "NONE",
    clone_target= "SELF",
    shared      = "yes"
]
```

1.9 Upgrading from OpenNebula 4.0.x

This guide describes the installation procedure for systems that are already running a 4.0.x OpenNebula. The upgrade will preserve all current users, hosts, resources and configurations; for both Sqlite and MySQL backends.

Read the Compatibility Guide for 4.2, 4.4 and *4.6*, and the Release Notes to know what is new in OpenNebula 4.6.

> **Warning:** With the new *multi-system DS* functionality, it is now required that the system DS is also part of the cluster. If you are using System DS 0 for Hosts inside a Cluster, any VM saved (stop, suspend, undeploy) **will not be able to be resumed after the upgrade process**.

> **Warning:** Two drivers available in 4.0 are now discontinued: **ganglia** and **iscsi**.
>   - **iscsi** drivers have been moved out of the main OpenNebula distribution and are available (although not supported) as an addon.
>   - **ganglia** drivers have been moved out of the main OpenNebula distribution and are available (although not supported) as an addon.

> **Warning:** There are combinations of **VMware storage** no longer supported (see *the VMFS Datastore guide* for the supported configurations).
> If you want to upgrade and you are using SSH, NFS or VMFS without SSH-mode, you will need to manually migrate your images to a newly created VMFS with SSH-mode datastore. To do so implies powering off all the VMs with images in any of the deprecated datastores, upgrade OpenNebula, create a VMFS datastore and then manually register the images from those deprecated datastores into the new one. Let us know if you have doubts or problems with this process.

1.9.1 Preparation

Before proceeding, make sure you don't have any VMs in a transient state (prolog, migr, epil, save). Wait until these VMs get to a final state (runn, suspended, stopped, done). Check the *Managing Virtual Machines guide* for more information on the VM life-cycle.

Stop OpenNebula and any other related services you may have running: EC2, OCCI, and Sunstone. As `oneadmin`, in the front-end:

```
$ sunstone-server stop
$ oneflow-server stop
$ econe-server stop
$ occi-server stop
$ one stop
```

1.9.2 Backup

Backup the configuration files located in **/etc/one**. You don't need to do a manual backup of your database, the onedb command will perform one automatically.

1.9.3 Installation

Follow the *Platform Notes* and the *Installation guide*, taking into account that you will already have configured the passwordless ssh access for oneadmin.

It is highly recommended **not to keep** your current `oned.conf`, and update the `oned.conf` file shipped with Open-Nebula 4.6 to your setup. If for any reason you plan to preserve your current `oned.conf` file, read the *Compatibility Guide* and the complete oned.conf reference for *4.0* and *4.6* versions.

1.9.4 Database Upgrade

The database schema and contents are incompatible between versions. The OpenNebula daemon checks the existing DB version, and will fail to start if the version found is not the one expected, with the message 'Database version mismatch'.

You can upgrade the existing DB with the 'onedb' command. You can specify any Sqlite or MySQL database. Check the *onedb reference* for more information.

> **Warning:** Make sure at this point that OpenNebula is not running. If you installed from packages, the service may have been started automatically.

After you install the latest OpenNebula, and fix any possible conflicts in oned.conf, you can issue the 'onedb upgrade -v' command. The connection parameters have to be supplied with the command line options, see the *onedb manpage* for more information. Some examples:

```
$ onedb upgrade -v --sqlite /var/lib/one/one.db
```

```
$ onedb upgrade -v -S localhost -u oneadmin -p oneadmin -d opennebula
```

If everything goes well, you should get an output similar to this one:

```
$ onedb upgrade -v -u oneadmin -d opennebula
MySQL Password:
Version read:
Shared tables 4.4.0 : OpenNebula 4.4.0 daemon bootstrap
Local tables  4.4.0 : OpenNebula 4.4.0 daemon bootstrap

>>> Running migrators for shared tables
  > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.0_to_4.4.1.rb
  > Done in 0.00s

  > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.1_to_4.5.80.rb
  > Done in 0.75s

Database migrated from 4.4.0 to 4.5.80 (OpenNebula 4.5.80) by onedb command.

>>> Running migrators for local tables
Database already uses version 4.5.80
Total time: 0.77s
```

If you receive the message "ATTENTION: manual intervention required", read the section *Manual Intervention Required* below.

> **Note:** Make sure you keep the backup file. If you face any issues, the onedb command can restore this backup, but it won't downgrade databases to previous versions.

1.9.5 Check DB Consistency

After the upgrade is completed, you should run the command `onedb fsck`.

First, move the 4.0 backup file created by the upgrade command to a safe place.

```
$ mv /var/lib/one/mysql_localhost_opennebula.sql /path/for/one-backups/
```

Then execute the following command:

```
$ onedb fsck -S localhost -u oneadmin -p oneadmin -d opennebula
MySQL dump stored in /var/lib/one/mysql_localhost_opennebula.sql
Use 'onedb restore' or restore the DB using the mysql command:
mysql -u user -h server -P port db_name < backup_file

Total errors found: 0
```

1.9.6 Update the Drivers

You should be able now to start OpenNebula as usual, running 'one start' as oneadmin. At this point, execute `onehost sync` to update the new drivers in the hosts.

> **Warning:** Doing `onehost sync` is important. If the monitorization drivers are not updated, the hosts will behave erratically.

1.9.7 Setting new System DS

With the new *multi-system DS* functionality, it is now required that the system DS is also part of the cluster. If you are using System DS 0 for Hosts inside a Cluster, any VM saved (stop, suspend, undeploy) **will not be able to be resumed after the upgrade process**.

You will need to have at least one system DS in each cluster. If you don't already, create new system DS with the same definition as the system DS 0 (TM_MAD driver). Depending on your setup this may or may not require additional configuration on the hosts.

You may also try to recover saved VMs (stop, suspend, undeploy) following the steps described in this thread of the users mailing list.

1.9.8 Testing

OpenNebula will continue the monitoring and management of your previous Hosts and VMs.

As a measure of caution, look for any error messages in oned.log, and check that all drivers are loaded successfully. After that, keep an eye on oned.log while you issue the onevm, onevnet, oneimage, oneuser, onehost **list** commands. Try also using the **show** subcommand for some resources.

1.9.9 Restoring the Previous Version

If for any reason you need to restore your previous OpenNebula, follow these steps:

- With OpenNebula 4.6 still installed, restore the DB backup using 'onedb restore -f'
- Uninstall OpenNebula 4.6, and install again your previous version.
- Copy back the backup of /etc/one you did to restore your configuration.

1.9.10 Known Issues

If the MySQL database password contains specials characters, such as @ or #, the onedb command will fail to connect to it.

The workaround is to temporarily change the oneadmin's password to an ASCII string. The set password statement can be used for this:

```
$ mysql -u oneadmin -p

mysql> SET PASSWORD = PASSWORD('newpass');
```

1.9.11 Manual Intervention Required

If you have a datastore configured to use a tm driver not included in the OpenNebula distribution, the onedb upgrade command will show you this message:

```
ATTENTION: manual intervention required

The Datastore <id> <name> is using the
custom TM MAD '<tm_mad>'. You will need to define new
configuration parameters in oned.conf for this driver, see
http://opennebula.org/documentation:rel4.4:upgrade
```

Since OpenNebula 4.4, each tm_mad driver has a TM_MAD_CONF section in oned.conf. If you developed the driver, it should be fairly easy to define the required information looking at the existing ones:

```
# The  configuration for each driver is defined in TM_MAD_CONF. These
# values are used when creating a new datastore and should not be modified
# since they define the datastore behaviour.
#   name      : name of the transfer driver, listed in the -d option of the
#               TM_MAD section
#   ln_target : determines how the persistent images will be cloned when
#               a new VM is instantiated.
#       NONE: The image will be linked and no more storage capacity will be used
#       SELF: The image will be cloned in the Images datastore
#       SYSTEM: The image will be cloned in the System datastore
#   clone_target : determines how the non persistent images will be
#                  cloned when a new VM is instantiated.
#       NONE: The image will be linked and no more storage capacity will be used
#       SELF: The image will be cloned in the Images datastore
#       SYSTEM: The image will be cloned in the System datastore
#   shared : determines if the storage holding the system datastore is shared
#            among the different hosts or not. Valid values: "yes" or "no"

TM_MAD_CONF = [
    name        = "lvm",
    ln_target   = "NONE",
    clone_target= "SELF",
    shared      = "yes"
]
```

1.10 Upgrading from OpenNebula 3.8.x

This guide describes the installation procedure for systems that are already running a 3.8.x OpenNebula. The upgrade will preserve all current users, hosts, resources and configurations; for both Sqlite and MySQL backends.

Read the Compatibility Guide for 4.0, 4.2, 4.4 and *4.6*, and the Release Notes to know what is new in OpenNebula 4.6.

> **Warning:** With the new *multi-system DS* functionality, it is now required that the system DS is also part of the cluster. If you are using System DS 0 for Hosts inside a Cluster, any VM saved (stop, suspend, undeploy) **will not be able to be resumed after the upgrade process**.

> **Warning:** Two drivers available in 3.8 are now discontinued: **ganglia** and **iscsi**.
>   • **iscsi** drivers have been moved out of the main OpenNebula distribution and are available (although not supported) as an addon.
>   • **ganglia** drivers have been moved out of the main OpenNebula distribution and are available (although not supported) as an addon.

> **Warning:** There are combinations of **VMware storage** no longer supported (see *the VMFS Datastore guide* for the supported configurations).
> If you want to upgrade and you are using SSH, NFS or VMFS without SSH-mode, you will need to manually migrate your images to a newly created VMFS with SSH-mode datastore. To do so implies powering off all the VMs with images in any of the deprecated datastores, upgrade OpenNebula, create a VMFS datastore and then manually register the images from those deprecated datastores into the new one. Let us know if you have doubts or problems with this process.

1.10.1 Preparation

Before proceeding, make sure you don't have any VMs in a transient state (prolog, migr, epil, save). Wait until these VMs get to a final state (runn, suspended, stopped, done). Check the *Managing Virtual Machines guide* for more information on the VM life-cycle.

Stop OpenNebula and any other related services you may have running: EC2, OCCI, and Sunstone. As `oneadmin`, in the front-end:

```
$ sunstone-server stop
$ oneflow-server stop
$ econe-server stop
$ occi-server stop
$ one stop
```

1.10.2 Backup

Backup the configuration files located in **/etc/one**. You don't need to do a manual backup of your database, the onedb command will perform one automatically.

1.10.3 Installation

Follow the *Platform Notes* and the *Installation guide*, taking into account that you will already have configured the passwordless ssh access for oneadmin.

It is highly recommended **not to keep** your current `oned.conf`, and update the `oned.conf` file shipped with OpenNebula 4.6 to your setup. If for any reason you plan to preserve your current `oned.conf` file, read the *Compatibility Guide* and the complete oned.conf reference for *3.8* and *4.6* versions.

1.10.4 Database Upgrade

The database schema and contents are incompatible between versions. The OpenNebula daemon checks the existing DB version, and will fail to start if the version found is not the one expected, with the message 'Database version mismatch'.

You can upgrade the existing DB with the 'onedb' command. You can specify any Sqlite or MySQL database. Check the *onedb reference* for more information.

> **Warning:** Make sure at this point that OpenNebula is not running. If you installed from packages, the service may have been started automatically.

After you install the latest OpenNebula, and fix any possible conflicts in oned.conf, you can issue the 'onedb upgrade -v' command. The connection parameters have to be supplied with the command line options, see the *onedb manpage* for more information. Some examples:

```
$ onedb upgrade -v --sqlite /var/lib/one/one.db
```

```
$ onedb upgrade -v -S localhost -u oneadmin -p oneadmin -d opennebula
```

If everything goes well, you should get an output similar to this one:

```
$ onedb upgrade -v -u oneadmin -d opennebula
MySQL Password:
Version read:
Shared tables 3.8.0 : OpenNebula 3.8.0 daemon bootstrap
Local tables  3.8.0 : OpenNebula 3.8.0 daemon bootstrap

MySQL dump stored in /var/lib/one/mysql_localhost_opennebula.sql
Use 'onedb restore' or restore the DB using the mysql command:
mysql -u user -h server -P port db_name < backup_file


>>> Running migrators for shared tables
  > Running migrator /usr/lib/one/ruby/onedb/shared/3.8.0_to_3.8.1.rb
  > Done in 0.36s

  > Running migrator /usr/lib/one/ruby/onedb/shared/3.8.1_to_3.8.2.rb
  > Done in 0.00s

  > Running migrator /usr/lib/one/ruby/onedb/shared/3.8.2_to_3.8.3.rb
  > Done in 0.00s

  > Running migrator /usr/lib/one/ruby/onedb/shared/3.8.3_to_3.8.4.rb
  > Done in 0.56s

  > Running migrator /usr/lib/one/ruby/onedb/shared/3.8.4_to_3.8.5.rb
  > Done in 0.00s

  > Running migrator /usr/lib/one/ruby/onedb/shared/3.8.5_to_3.9.80.rb

ATTENTION: manual intervention required
Virtual Machine deployment files have been moved from /var/lib/one to
/var/lib/one/vms. You need to move these files manually:

    $ mv /var/lib/one/[0-9]* /var/lib/one/vms

  > Done in 1.10s

  > Running migrator /usr/lib/one/ruby/onedb/shared/3.9.80_to_3.9.90.rb

ATTENTION: manual intervention required
IM and VM MADS have been renamed in oned.conf. To keep your
existing hosts working, you need to duplicate the drivers with the
old names.

For example, for kvm you will have IM_MAD "kvm" and VM_MAD "kvm", so you
```

```
need to add IM_MAD "im_kvm" and VM_MAD "vmm_kvm"

IM_MAD = [
     name       = "kvm",
     executable = "one_im_ssh",
     arguments  = "-r 0 -t 15 kvm" ]


IM_MAD = [
     name       = "im_kvm",
     executable = "one_im_ssh",
     arguments  = "-r 0 -t 15 kvm" ]

VM_MAD = [
    name       = "kvm",
    executable = "one_vmm_exec",
    arguments  = "-t 15 -r 0 kvm",
    default    = "vmm_exec/vmm_exec_kvm.conf",
    type       = "kvm" ]

VM_MAD = [
    name       = "vmm_kvm",
    executable = "one_vmm_exec",
    arguments  = "-t 15 -r 0 kvm",
    default    = "vmm_exec/vmm_exec_kvm.conf",
    type       = "kvm" ]

 > Done in 0.41s

 > Running migrator /usr/lib/one/ruby/onedb/shared/3.9.90_to_4.0.0.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.0.0_to_4.0.1.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.0.1_to_4.1.80.rb
 > Done in 0.09s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.1.80_to_4.2.0.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.2.0_to_4.3.80.rb
 > Done in 0.68s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.3.80_to_4.3.85.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.3.85_to_4.3.90.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.3.90_to_4.4.0.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.0_to_4.4.1.rb
 > Done in 0.00s

 > Running migrator /usr/lib/one/ruby/onedb/shared/4.4.1_to_4.5.80.rb
 > Done in 0.39s
```

```
Database migrated from 3.8.0 to 4.5.80 (OpenNebula 4.5.80) by onedb command.

>>> Running migrators for local tables
Database already uses version 4.5.80

Total time: 3.60s
```

Note: Make sure you keep the backup file. If you face any issues, the onedb command can restore this backup, but it won't downgrade databases to previous versions.

1.10.5 Check DB Consistency

After the upgrade is completed, you should run the command `onedb fsck`.

First, move the 4.0 backup file created by the upgrade command to a safe place.

```
$ mv /var/lib/one/mysql_localhost_opennebula.sql /path/for/one-backups/
```

Then execute the following command:

```
$ onedb fsck -S localhost -u oneadmin -p oneadmin -d opennebula
MySQL dump stored in /var/lib/one/mysql_localhost_opennebula.sql
Use 'onedb restore' or restore the DB using the mysql command:
mysql -u user -h server -P port db_name < backup_file

Total errors found: 0
```

1.10.6 Virtual Machine Directories

Note: Only for OpenNebula versions < 3.8.3

If you are upgrading from a version **lower than 3.8.3**, you need to move the Virtual Machine deployment files from '/var/lib/one' to '/var/lib/one/vms':

```
$ mv /var/lib/one/[0-9]* /var/lib/one/vms
```

1.10.7 Driver Names

OpenNebula default driver names have changed in the configuration file. Now the names of the vmm and im drivers are not prepended by the type of driver:

- vmm_kvm → kvm
- vmm_xen → xen
- vmm_vmware → vmware
- vmm_ec2 → ec2
- vmm_dummy → dummy
- im_kvm → kvm
- im_xen → xen

- im_vmware → vmware

- im_ec2 → ec2

- im_ganglia → ganglia

- im_dummy → dummy

To keep your existing hosts working, you need to duplicate the drivers with the old names.

For example, for kvm you will have IM_MAD `kvm` and VM_MAD `kvm`, so you need to add IM_MAD `im_kvm` and VM_MAD `vmm_kvm`

```
IM_MAD = [
    name       = "kvm",
    executable = "one_im_ssh",
    arguments  = "-r 3 -t 15 kvm" ]

IM_MAD = [
    name       = "im_kvm",
    executable = "one_im_ssh",
    arguments  = "-r 3 -t 15 kvm" ]

VM_MAD = [
    name       = "kvm",
    executable = "one_vmm_exec",
    arguments  = "-t 15 -r 0 kvm",
    default    = "vmm_exec/vmm_exec_kvm.conf",
    type       = "kvm" ]

VM_MAD = [
    name       = "vmm_kvm",
    executable = "one_vmm_exec",
    arguments  = "-t 15 -r 0 kvm",
    default    = "vmm_exec/vmm_exec_kvm.conf",
    type       = "kvm" ]
```

1.10.8 Manual Intervention Required

Note: Ignore this section if onedb didn't output the following message

If you have a datastore configured to use a tm driver not included in the OpenNebula distribution, the onedb upgrade command will show you this message:

```
ATTENTION: manual intervention required

The Datastore <id> <name> is using the
custom TM MAD '<tm_mad>'. You will need to define new
configuration parameters in oned.conf for this driver, see
http://opennebula.org/documentation:rel4.4:upgrade
```

Since OpenNebula 4.4, each tm_mad driver has a TM_MAD_CONF section in oned.conf. If you developed the driver, it should be fairly easy to define the required information looking at the existing ones:

```
# The  configuration for each driver is defined in TM_MAD_CONF. These
# values are used when creating a new datastore and should not be modified
# since they define the datastore behaviour.
#   name       : name of the transfer driver, listed in the -d option of the
```

```
#                TM_MAD section
#   ln_target : determines how the persistent images will be cloned when
#               a new VM is instantiated.
#       NONE: The image will be linked and no more storage capacity will be used
#       SELF: The image will be cloned in the Images datastore
#       SYSTEM: The image will be cloned in the System datastore
#   clone_target : determines how the non persistent images will be
#                  cloned when a new VM is instantiated.
#       NONE: The image will be linked and no more storage capacity will be used
#       SELF: The image will be cloned in the Images datastore
#       SYSTEM: The image will be cloned in the System datastore
#   shared : determines if the storage holding the system datastore is shared
#            among the different hosts or not. Valid values: "yes" or "no"

TM_MAD_CONF = [
    name        = "lvm",
    ln_target   = "NONE",
    clone_target= "SELF",
    shared      = "yes"
]
```

1.10.9 Update the Drivers

You should be able now to start OpenNebula as usual, running 'one start' as oneadmin. At this point, execute `onehost sync` to update the new drivers in the hosts.

> **Warning:** Doing `onehost sync` is important. If the monitorization drivers are not updated, the hosts will behave erratically.

1.10.10 Setting new System DS

With the new *multi-system DS* functionality, it is now required that the system DS is also part of the cluster. If you are using System DS 0 for Hosts inside a Cluster, any VM saved (stop, suspend, undeploy) **will not be able to be resumed after the upgrade process**.

You will need to have at least one system DS in each cluster. If you don't already, create new system DS with the same definition as the system DS 0 (TM_MAD driver). Depending on your setup this may or may not require additional configuration on the hosts.

You may also try to recover saved VMs (stop, suspend, undeploy) following the steps described in this thread of the users mailing list.

1.10.11 Testing

OpenNebula will continue the monitoring and management of your previous Hosts and VMs.

As a measure of caution, look for any error messages in oned.log, and check that all drivers are loaded successfully. After that, keep an eye on oned.log while you issue the onevm, onevnet, oneimage, oneuser, onehost **list** commands. Try also using the **show** subcommand for some resources.

1.10.12 Restoring the Previous Version

If for any reason you need to restore your previous OpenNebula, follow these steps:

- With OpenNebula 4.6 still installed, restore the DB backup using 'onedb restore -f'

- Uninstall OpenNebula 4.6, and install again your previous version.

- Copy back the backup of /etc/one you did to restore your configuration.

1.10.13 Known Issues

If the MySQL database password contains specials characters, such as @ or #, the onedb command will fail to connect to it.

The workaround is to temporarily change the oneadmin's password to an ASCII string. The set password statement can be used for this:

```
$ mysql -u oneadmin -p

mysql> SET PASSWORD = PASSWORD('newpass');
```