

D3.1.1	
Version	1.0
Author	USV
Dissemination	PU
Date	22/01/2014
Status	Final



D3.1.1: NUBOMEDIA virtual infrastructure v1

Project acronym:	NUBOMEDIA
Project title:	NUBOMEDIA: an elastic Platform as a Service (PaaS) cloud for interactive social multimedia
Project duration:	2014-02-01 to 2016-09-30
Project type:	STREP
Project reference:	610576
Project web page:	http://www.nubomedia.eu
Work package	WP3: NUBOMEDIA cloud platform
WP leader	Giuseppe Carella (TUB)
Deliverable nature:	Prototype
Lead editor:	Cristian Spoiala (USV)
Planned delivery date	01/2015
Actual delivery date	22/01/2015
Keywords	Virtual infrastructure, OpenStack, API capabilities, Docker

The research leading to these results has been funded by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 610576



FP7 ICT-2013.1.6. Connected and Social Media



This is a public deliverable that is provided to the community under a **Creative Commons Attribution-ShareAlike 4.0 International** License

<http://creativecommons.org/licenses/by-sa/4.0/>

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material
for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

For a full description of the license legal terms, please refer to:

<http://creativecommons.org/licenses/by-sa/4.0/legalcode>

Contributors:

Cristian Spoiala (USV)
Alin Calinciuc (USV)
Constantin Filote (USV)

Internal Reviewer(s):

Giuseppe Carella (TUB)
Lorenzo Tomasini (TUB)
Luis López (URJC)
Javier López (NAEVATEC)

Version History

Version	Date	Authors	Comments
0.1	16/07/2014	Cristian Spoiala (USV), Alin Calinciuc (USV), Constantin Filote (USV)	Initial Version
0.2	26/09/2014	Cristian Spoiala (USV), Alin Calinciuc (USV), Constantin Filote (USV)	Added Docker and Neutron API sections
0.3	30/09/2014	Cristian Spoiala (USV), Alin Calinciuc (USV), Constantin Filote (USV)	Added changes from Lorenzo review
0.4	3/11/2014	Cristian Spoiala (USV), Alin Calinciuc (USV), Constantin Filote (USV)	Updated version history and reviewers. Added section describing our specific OpenStack
0.5	18/12/2014	Cristian Spoiala (USV), Alin Calinciuc (USV),	Added NUBOMEDIA R3 changes
1.0	11/01/2014	Alin Calinciuc (USV)	Added more details about NUBOMEDIA infrastructure in relation with Openstack

Table of contents

1	Executive summary	9
2	State-of-the-art	9
2.1	OpenStack	9
2.1.1	Nova	9
2.1.2	Keystone	10
2.1.3	Glance	11
2.1.4	Neutron	11
2.1.5	Telemetry (Ceilometer)	12
2.1.6	Cinder	14
2.1.7	Heat	14
2.1.8	Oslo	14
2.1.9	Horizon	14
2.2	OpenNebula	14
2.3	CloudStack	14
2.4	Docker	15
2.4.1	Overview	15
2.4.2	Architecture	15
2.4.3	Docker features:	16
2.4.4	Docker components:	16
2.4.5	Docker use cases	17
2.4.6	Issues	17
2.4.7	New Docker tools	17
3	NUBOMEDIA infrastructure	18
3.1	Why OpenStack	18
3.2	OpenStack components used	19
3.2.1	Current OpenStack configuration	19
3.2.2	Current OpenStack R2 configuration	19
3.2.3	Current OpenStack R3 configuration	20
3.3	Docker	21
4	NUBOMEDIA API capabilities	22
4.1	Programming languages support	22
4.1.1	SDK	22
4.1.2	REST API	23
5	NUBOMEDIA API examples	23
5.1	Compute API	23
5.1.1	Authenticate and request token	23
5.1.2	Lists IDs, names, and links for all servers	27
5.1.3	Gets details for a specified server	27
5.1.4	Lists networks and addresses for a specified tenant and server	29
5.1.5	Hard reboot instance	29
5.1.6	Lists all details for available flavors	29
5.1.7	Create Instance	31
5.2	Neutron API	32
5.2.1	Authentication and authorization	32
5.2.2	Filtering the response and column selection	33
5.2.3	Neutron asynchronous behavior	34
5.3	Telemetry API	34
5.3.1	Meters	35
6	References	36

List of Figures:

Figure 1 OpenStack Architecture 13

Figure 2 Container vs VMs architecture comparison..... 16

Figure 3 NUBOMEDIA OpenStack cloud infrastructure architecture 20

Figure 4 OpenStack with docker hypervisor architecture 22

Acronyms and abbreviations:

SDN	Software-defined networking
IaaS	Infrastructure as a service
PaaS	Platform as a service
DVR	Distributed Virtual Router
LBaaS	Load-Balancing-as-a-Service

1 Executive summary

This document provides the appropriate software artifacts prototyping the NUBOMEDIA cloud infrastructure including all software components and documentation on how each of them is used to accomplish needed tasks on the NUBOMEDIA platform.

2 State-of-the-art

2.1 OpenStack

OpenStack is a free and open-source software cloud computing platform. It is primarily deployed as an infrastructure as a service (IaaS) solution. The technology consists of a series of interrelated projects that control pools of processing, storage, and networking resources throughout a data center, able to be managed or provisioned through a web-based dashboard, command-line tools, a RESTful API, or SDKs. It is released under the terms of the Apache License.

Its main services are Nova, Keystone, Glance, Neutron, Cinder, Ceilometer, Heat, Oslo and Horizon. This document presents all these services and their responsibility on OpenStack infrastructure.

2.1.1 Nova

Nova is the project name for OpenStack Compute, a cloud computing fabric controller, the main part of an IaaS system. Individuals and organizations can use Nova to host and manage their own cloud computing systems. Nova originated as a project of NASA Ames Research Laboratory.

Nova is written with the following design guidelines in mind:

- Component based architecture: Quickly add new behaviors
- Highly available: Scale to very serious workloads
- Fault-Tolerant: Isolated processes avoid cascading failures
- Recoverable: Failures should be easy to diagnose, debug, and rectify
- Open Standards: Be a reference implementation for a community-driven API
- API Compatibility: Nova strives to provide API-compatible with popular systems like Amazon EC2

OpenStack Nova provides a cloud computing fabric controller, supporting a wide variety of virtualization technologies, including KVM, Xen, LXC, VMware, and more. In addition to its native API, it includes compatibility with the commonly encountered Amazon EC2 and S3 APIs.

Nova primarily consists of a set of Python daemons, though it requires and integrates with a number of native system components for databases, messaging, and virtualization capabilities.

[Here](#) you can find a list of all supported hypervisors that can be used with Nova deployment. Not all drivers are fully supported and not all of them support the same features.

Here we try to categorize them by their testing and development status, so we have:

1. Group A

These drivers are fully supported. Test coverage includes:

- unit tests that gate commits
- functional testing that gate commits

Drivers in this group include:

- libvirt (qemu/KVM on x86)

2. Group B

These drivers are in a bit of a middle ground. Test coverage includes:

- unit tests that gate commits
- functional testing providing by an external system that does not gate commits, but advises patch authors and reviewers of results in gerrit (the code review system).

Drivers in this group include:

- Hyper-V
- VMware
- XenServer

3. Group C

Drivers in Group C are deprecated. These drivers have minimal testing, and may or may not work at any given time. Use them at your own risk.

Test coverage includes:

- (maybe) unit tests that gate commits
- no public functional testing

Drivers in this group include.

- baremetal
- Docker
- Xen via libvirt
- LXC via libvirt

2.1.2 Keystone

Keystone provides authentication, authorization, and service discovery mechanisms via HTTP primarily for use by projects in the OpenStack family. It is most commonly deployed as an HTTP interface to existing identity systems, such as LDAP.

It supports multiple forms of authentication including standard username and password credentials, token-based systems and AWS-style (i.e. Amazon Web Services) logins. Additionally, the catalog provides a queryable list of all of the services deployed in an OpenStack cloud in a single registry. Users and third-party tools can programmatically determine which resources they can access.

Keystone is organized as a group of internal services exposed on one or many endpoints. Many of these services are used in a combined fashion by the frontend, for example an authenticate call will validate user/tenant credentials with the Identity service and, upon success, create and return a token with the Token service.

2.1.3 Glance

The Glance project provides services for discovering, registering, and retrieving virtual machine images. Glance has a RESTful API that allows querying of VM image metadata as well as retrieval of the actual image.

VM images made available through Glance can be stored in a variety of locations from simple filesystems to object-storage systems like the OpenStack Swift project.

2.1.4 Neutron

The Networking service, code-named neutron, provides an API that lets you define network connectivity and addressing in the cloud. The Networking service enables operators to leverage different networking technologies to power their cloud networking. The Networking service also provides an API to configure and manage a variety of network services ranging from L3 forwarding and NAT to load balancing, edge firewalls, and IPSEC VPN.

OpenStack Networking ships with plug-ins and agents for Cisco virtual and physical switches, NEC OpenFlow products, Open vSwitch, Linux bridging, Ryu Network Operating System, and the VMware NSX product.

Using the Neutron API users are able to list, show information for, create, update, and delete networks and subnet resources. Also you are able to manage security groups and rules (security-groups) by listing, creating, editing and deleting them.

At Layer 3 routers route packets between subnets, forward packets from internal networks to external ones, and access instances from external networks through floating IPs.

Neutron introduces these resources:

- router. A logical entity for forwarding packets across internal subnets and NATting them on external networks through an appropriate external gateway.
- floatingip. An external IP address that is mapped to a port that is attached to an internal network.

The common agents that neutron uses are based on L3 (layer 3), DHCP (dynamic host IP addressing), and a plug-in agent.

Neutron mission statement is to implement services and associated libraries to provide on-demand, scalable, and technology-agnostic network abstraction.

Neutron is an OpenStack project to provide "networking as a service" between interface devices (e.g., vNICs) managed by other OpenStack services (e.g., nova).

Networking is a standalone component in the OpenStack modular architecture. It's positioned alongside OpenStack components such as Compute, Image Service, Identity, or the Dashboard. Like those components, a deployment of Networking often involves deploying several services to a variety of hosts.

The Networking server uses the neutron-server daemon to expose the Networking API and enable administration of the configured Networking plug-in. Typically, the plug-in

requires access to a database for persistent storage (also similar to other OpenStack services).

If your deployment uses a controller host to run centralized Compute components, you can deploy the Networking server to that same host. However, Networking is entirely standalone and can be deployed to a dedicated host. Depending on your configuration, Networking can also include the following agents:

- plug-in agent
- dhcp agent
- l3 agent
- metering agent

These agents interact with the main neutron process through RPC (for example, RabbitMQ or Qpid) or through the standard Networking API. In addition, Networking integrates with OpenStack components in a number of ways:

Networking relies on the Identity service (keystone) for the authentication and authorization of all API requests.

Compute (nova) interacts with Networking through calls to its standard API. As part of creating a VM, the nova-compute service communicates with the Networking API to plug each virtual NIC on the VM into a particular network.

The dashboard (horizon) integrates with the Networking API, enabling administrators and tenant users to create and manage network services through a web-based GUI.

2.1.5 Telemetry (Ceilometer)

The Telemetry (Ceilometer) project aims to deliver a unique point of contact for billing systems to acquire all of the measurements they need to establish customer billing, across all current OpenStack core components with work underway to support future OpenStack components.

Ceilometer was the initial name and project purpose was to do metering. Starting with Havana release, the project was renamed Telemetry, and also alarming feature was added.

Ceilometer can publish information for monitoring, debugging and graphing tools in addition or in parallel to the metering backend.

2.1.5.1 Features

- API
- Multiple storage support: MongoDB (default), SQL, HBase
- Alarming
- Metering
- Multi-publisher

2.1.5.2 Architecture

The figure is a representation of how the collectors and agents gather data from multiple sources.

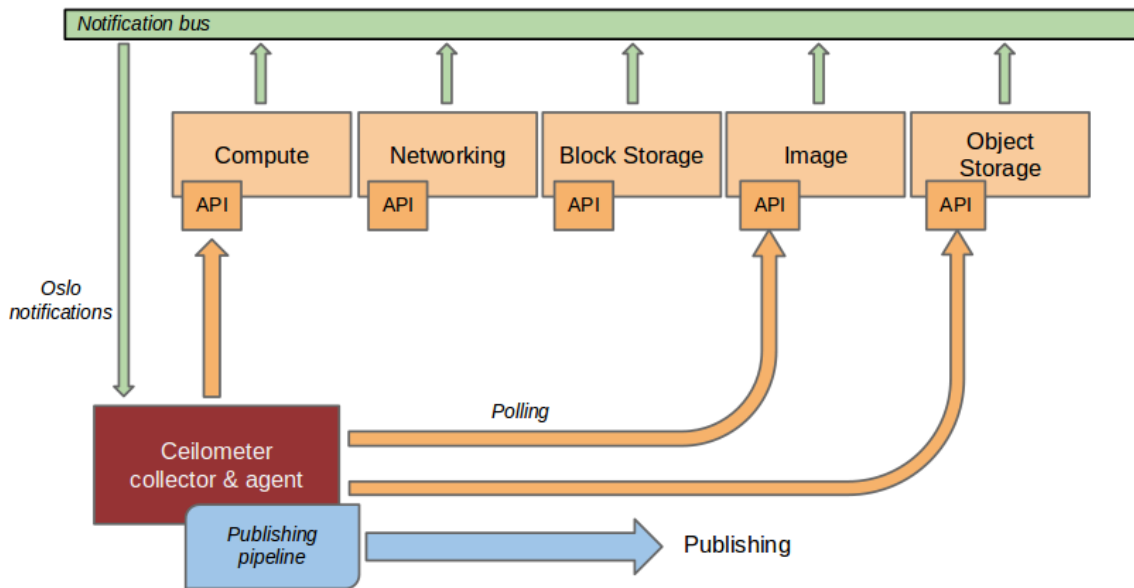


Figure 1 OpenStack Architecture

Ceilometer project created 3 independent methods to collect data:

- **Bus listener agent** which takes events generated on the Oslo notification bus and transforms them into Ceilometer samples. This is the preferred method of data collection.
- **Push agents** which is the only solution to fetch data within projects, which do not expose the required data in a remotely usable way. This is not the preferred method as it makes deployment a bit more complex having to add a component to each of the nodes that need to be monitored. However, we do prefer this compared to a polling agent method since resilience (high availability) will not be a problem with this method.
- 1. **Polling agents** which is the least preferred method, that will poll some API or other tool to collect information at a regular interval. This method is least preferred due to the inherent difficulty in making such a component resilient.

The Alarming component of Ceilometer, first delivered in the Havana version, allows you to set alarms based on threshold evaluation for a collection of samples. An alarm can be set on a single meter, or on a combination. For example, you may want to trigger an alarm when the memory consumption reaches 70% on a given instance if the instance has been up for more than 10 min. To setup an alarm, you will call Ceilometer's API server specifying the alarm conditions and an action to take.

There can be multiple forms of actions, but two have been implemented so far:

- **HTTP callback:** you provide a URL to be called whenever the alarm has been set off. The payload of the request contains all the details of why the alarm was triggered.

- **Log:** mostly useful for debugging, stores alarms in a log file.

2.1.6 Cinder

Cinder is the Block Storage service on OpenStack. It's designed to allow the use of a reference implementation (LVM) to present storage resources to end users that can be consumed by the OpenStack Nova.

As cinder storage users can configure a NFS share, a GlusterFS server, a Ceph server, or a dedicate storage platform like IBM Storage, NetAPP, Nexenta, SolidFire, etc.

2.1.7 Heat

OpenStack Heat is the OpenStack orchestration program that is made to create a human and machine-accessible service for managing the entire lifecycle of infrastructure and applications within OpenStack clouds.

Heat is made to give a cloud administrator the ability to easy launch (orchestrate) multiple composite cloud applications based on templates that can be different text format files. Heat has a native template format which can be used but it is still not very mature, but it also supports other orchestration template systems like [AWS CloudFormation](#) template, and is also has a [CloudWatch](#) query API.

Heat manages the whole lifecycle of the application. When you need to change you infrastructure you can only modify you template and use it to update your existing stack.

2.1.8 Oslo

The Oslo project produces a set of python libraries containing infrastructure code shared by OpenStack projects. The APIs provided by these libraries should be high quality, stable, consistent and generally useful.

2.1.9 Horizon

Horizon is the OpenStack dashboard. It provides a web based user interface to manage OpenStack services including Nova, Heat, Glance, Cinder, Swift, Keystone and others. It integrates only basic things that OpenStack components can do, for advanced things OpenStack administrators can use python-clients to do much more than can be done on Horizon.

2.2 OpenNebula

OpenNebula is a cloud computing toolkit for managing heterogeneous distributed data center infrastructures. The OpenNebula toolkit manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. OpenNebula is free and open-source software, subject to the requirements of the Apache License version 2.

2.3 CloudStack

CloudStack is open source cloud computing software for creating, managing, and deploying infrastructure cloud services. It uses existing hypervisors such as KVM, vSphere, and XenServer/XCP for virtualization. In addition to its own API, CloudStack also supports the Amazon Web Services (AWS) API, and the Open Cloud Computing Interface from the Open Grid Forum.

2.4 Docker

2.4.1 Overview

Docker is an open-source project that automates the deployment of applications inside self-sufficient software containers. It is mainly targeted at developers and system administrators to build, ship, and run distributed applications.

Containers are independent of hardware, language, framework, and hosting provider. To achieve its goals, Docker is using kernel namespacing, cgroups, and LXC with high level API. It provides a way to automate software development in a secure and repeatable environment.

Docker is a way to manage LXC containers on a single machine.

At this moment, Docker can run on any x86 Linux Kernel that supports cgroups and aufs, and it aims for full OpenStack compatibility.

Containers existed before Docker, but they were not standardized and not easy to use. The aim of Docker is to facilitate the use of containers and to make their use convenient on any platform.

2.4.2 Architecture

Virtual machines run virtually on physical hardware via an intermediation layer, while containers run inside user space on top of an operating system kernel (currently Linux). This allows running multiple isolated user spaces on a single host.

Containers vs. VMs

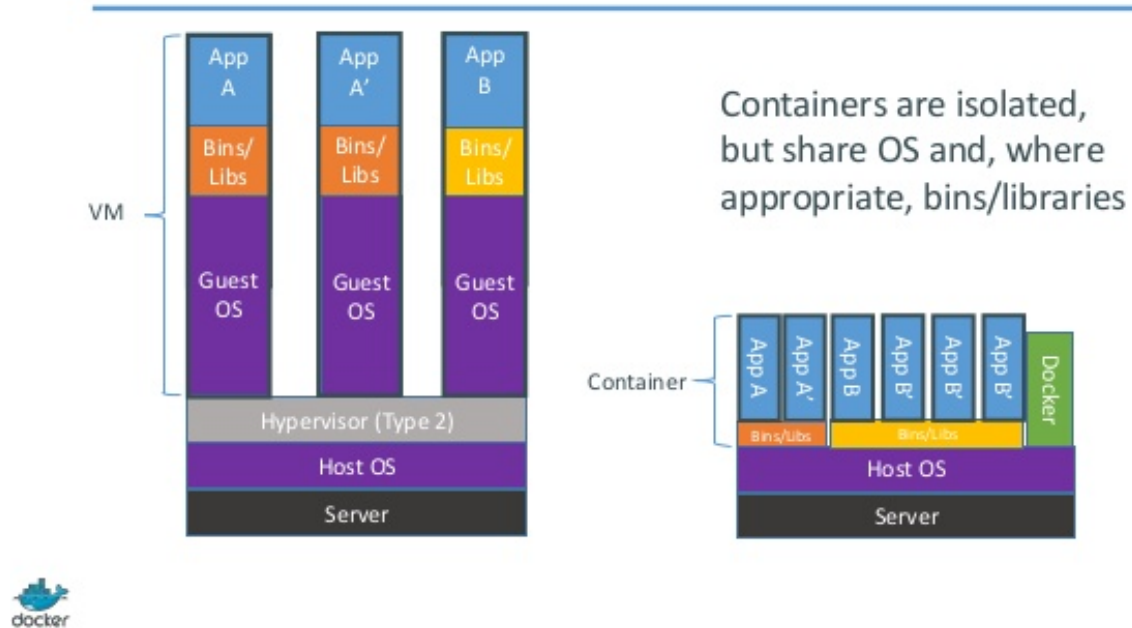


Figure 2 Container vs VMs architecture comparison

2.4.3 Docker features:

- Speed (containers are started in milliseconds);
- Developers focus on code and not on operations;
- Portable deployment across machines;
- Component re-use;
- Versioning of builds;
- Application centric.

2.4.4 Docker components:

- Docker client and server;
- Docker images;
- Registries;
- Docker containers.

2.4.4.1 Docker client and server

Docker is a client-server application where client is talking to the server which runs the containers.

2.4.4.2 Docker images

Containers are started from images, and they are the starting source code on top of which the container will be built. They use Union file system.

2.4.4.3 Registries (DockerHub)

Docker stores the images that were built on registries. Registries can be public or private. Company behind Docker operates a service to store images called DockerHub. Furthermore, private registries can be stored freely for your organization.

2.4.4.4 Containers

Docker helps you build and deploy containers inside which you can package your applications and services. Containers are launched from images, and can contain one or more processes.

A Docker container is:

- An image format;
- A set of standard operations;
- An execution environment.

A container can be any piece of software, ranging from a web server to a NoSQL database.

2.4.5 Docker use cases

Docker and containers can be used for many use cases. Some of them:

- Speeding up software development by improving the test-staging-production workflow by building efficient and lightweight containers. Same containers can be tested, then pushed to staging, and finally deployed on production;
- Building a multi-user PaaS infrastructure;
- Run small services like Memcached, Redis, and build a SaaS;
- Continuous integration systems.

2.4.6 Issues

2.4.6.1 Security

Considering the level of isolation and attack vectors, security for Docker requires respecting a list of best practices.

To secure a Docker environment, users have to use SELinux or AppArmor, secure kernel with gsec, read-only mounts. Do not use root for apps.

2.4.7 New Docker tools

Open source community did not end with Docker since new tools in development will offer new functionality.

2.4.7.1 CoreOS

CoreOS is a Linux distribution that uses Linux containers (Docker) to manage services and clusters of servers. Cloud providers like Google and Digital Ocean have announced official support for CoreOS images.

2.4.7.2 *Atomic Project*

Atomic Project, which is sponsored by RedHat is similar with CoreOS, but it is mostly aimed for enterprise users, and oriented around RedHat ecosystem (RedHat Enterprise Linux, Fedora, CentOS). It provides an end-to-end solution around Docker and applications.

2.4.7.3 *Kubernetes*

Google had open sourced its tools for Docker cluster management. Companies like Microsoft, RedHat, IBM have already announced their support for this open container framework, and they will work closely to support common tools and avoid vendor lock-in.

3 NUBOMEDIA infrastructure

3.1 Why OpenStack

In order to fulfill the virtualization needs of the NUBOMEDIA project, a stable open source platform with all capabilities supported was required. From all open source IaaS platforms, OpenStack is the most feature complete and mature, and it is used in production by many large companies like HP, Rackspace, Cisco, Paypal. It also solves one of the needs of the NUBOMEDIA project to provide a stable virtual infrastructure for project to build a low latency multimedia traffic platform.

All capabilities needed in the project are fulfilled by OpenStack project. Capabilities like:

- Support for Xen and KVM
- Storage planning and management (Swift or Gluster are supported)
- Support for different configuration and schemes, SDN (fulfilled with Neutron)
- Support to gather performance metrics (fulfilled by Ceilometer)

On top of OpenStack capabilities, there is a well-documented API that enables NUBOMEDIA project to build an elastic cloud platform. OpenStack performs critical operations like instance creation, gathering performance metrics, and orchestration of components very reliably.

OpenStack also use commodity hardware that allowed us to have a testbed from multiple nodes and also enable future users of the NUBOMEDIA platform to have flexibility in the hardware requirements.

Also support for critical components like Networking Layer (Neutron) is one of the best in industry with support for all major vendors like Cisco, Juniper, Mellanox, NEC, etc.

For NUBOMEDIA R2 we have chosen Open vSwitch because it is a production quality, multilayer virtual switch licensed under the open source Apache 2.0 license.

In addition, the APIs available on OpenStack are well documented and stable to build a flexible platform.

3.2 OpenStack components used

In NUBOMEDIA we use the following OpenStack components:

- **Nova** for virtual machines
- **Neutron** for networking
- **Ceilometer** for metrics and alerts
- **Glance** for images
- **Keystone** for authentication
- **Heat** for orchestration
- **Horizon** for web admin dashboard

3.2.1 Current OpenStack configuration

3.2.2 Current OpenStack R2 configuration

For NUBOMEDIA's R2 release we installed OpenStack Icehouse from RedHat. As host operating system we used Centos 6.5 with some fine tuning in order to better support OpenStack (more information available on NUBOMEDIA's deliverable D6.1.1).

On top of the standard installation of these packages we made some modifications to fulfill NUBOMEDIA needs:

- We changed the interval that defines how often ceilometer is gathering data from virtual machines on `/etc/ceilometer/pipeline.yaml`. This setting must be done on all compute instances running nova-compute. We change the interval because we need the heat orchestrator have cpu utilization information every minute so it can take decisions based on the load of each instance. The source code can be found [here](#).
- We also created a script to release all unused floating ips and created a cronjob that runs each 5 minutes cleaning all floating ips. We configured this script in order to free all unused public ip address so they can be provisioned and used again when running a heat template. You can find the script that needs to be configured as cronjob [here](#).
- We configured nova to remove all unused _base images in order to preserve free space. On `/etc/nova/nova.conf` we configured:
`image_cache_manager_interval=60`
`remove_unused_base_images=true`
`remove_unused_original_minimum_age_seconds=300`

On R2 we considered that QEMU/KVM hypervisor because KVM is the most developed and up to date hypervisor today and it provides integration with all other

OpenStack features like Inject Networking, Live migration, VLAN and VXLAN networking, firewall rules.

Currently on R2 we have configured a cluster that is composed of 5 OpenStack compute nodes that have 2 x Quad-Core AMD Opteron(tm) Processor 2376 HE with 16Gb of RAM, 1 SCSI HDD of 137GB and two gigabit network ports and one infinity band. All hosts are interconnected using a full gigabit management switch. The connectivity between compute nodes and the IBM where all glance images and cinder volumes are kept. For network traffic we use tagged VLANs in order to assure a good isolation between internal OpenStack instances and the rest of the network.

We have also setup a pool of 32 public IP addresses (80.96.122.48-80.96.122.80) that are used by instances in OpenStack.

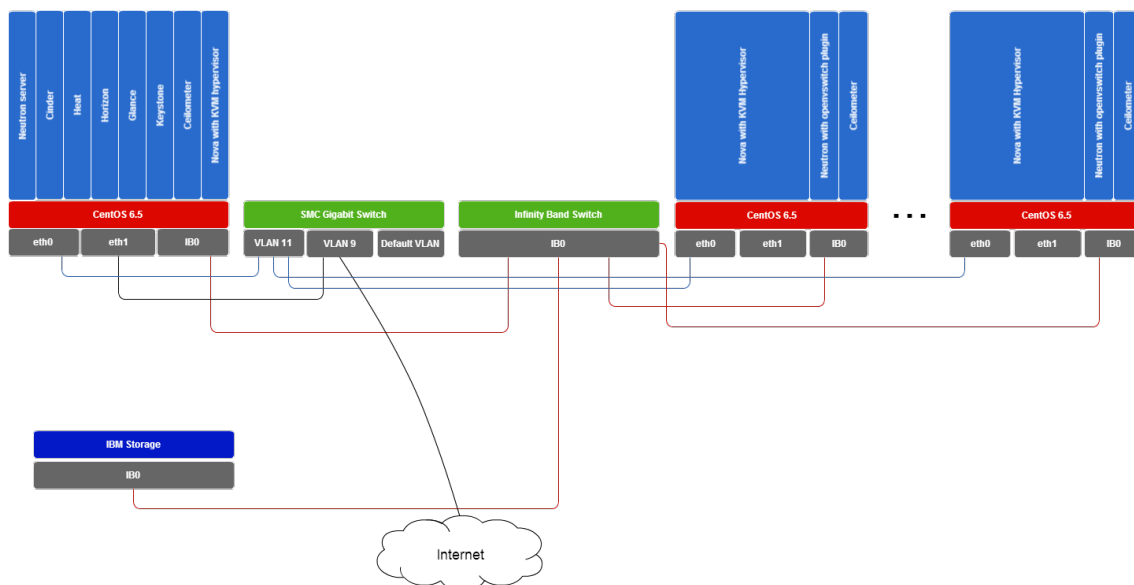


Figure 3 NUBOMEDIA OpenStack cloud infrastructure architecture

We use VLAN 11 for internal traffic between all compute nodes and the master node and all compute nodes, when they are assigned a floating IP the whole traffic is directed through the neutron server from the master node. VLAN 9 is used on eth1 of the compute node in order to all instances to access the internet. Openvswitch from the master node is routing all traffic from OpenStack intra-network on eth0 to eth1.

3.2.3 Current OpenStack R3 configuration

For NUBOMEDIA release 3 we installed OpenStack Juno from RedHat. The Juno version from RDO currently only supports RHEL 7 and CentOS 7 operating systems. The main advantages of Juno release is the stability of the ML2 plugin, the great support for VXLANs and the DVR capability.

Modular Layer 2 (ML2) is a neutron plugin allowing OpenStack networking to simultaneously utilize the variety of layer 2 network technologies like VLAN, VXLAN, Flat, and GRE.

Virtual Extensible LAN (VXLAN) is a network virtualization technology that attempts to ameliorate the scalability problems associated with large cloud computing deployments. It uses a VLAN-like encapsulation technique to encapsulate MAC-based OSI layer 2 Ethernet frames within layer 4 UDP packets.

With DVR (Distributed Virtual Routing), L3 forwarding and NAT are now distributed to the compute nodes. This means that with DVR every compute node needs to act as a network node, that provides both L3 and NAT. So now, if two instances on the same physical server need to communicate between them they will not need to use the router on the "network node", they will be able to communicate directly. Also if a instance from a compute node needs to access the external network it can do that by using it's router and not making traffic on the private network between compute nodes.

Although DVR is not a stable feature of OpenStack Juno we managed to install and test it's performance by configuring it in the following way:

1. First you should install OpenStack Juno using the steps provided on NUBOMEDIA's D6.1.1 document. In the configurations of the packstack answer file you should configure the L2 to use the ML2 plugin, and the tenant network type should be VXLAN.
2. After the install is done and you made all other necessary adjustments you should change the neutron configuration in the following way:
 1. On the maste node on `/etc/neutron/l3_agent.ini` you should change `agent_mode = dvr_snat`.
 2. On compute nodes you should change the `agent_mode = dvr` on `/etc/neutron/l3_agent.ini` file.
 3. On master node you should configure on `/etc/neutron/neutron.conf` file the `router_distributed = True`.
 4. On `/etc/neutron/plugins/ml2/ml2_conf.ini` you should append `l2population` to `mechanism_drivers`.
 5. On each compute and master node file `/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini` you should configure `l2_population = True` , `tunnel_types = vxlan` and `enable_distributed_routing = True`.
3. You shold update the neutron database with the new configurations running the following bash commands:


```
mysql -e "drop database if exists neutron_ml2;"
mysql -e "create database neutron_ml2 character set utf8;"
mysql -e "grant all on neutron_ml2.* to 'neutron'('@'%);"
neutron-db-manage --config-file /usr/share/neutron/neutron-dist.conf --
config-file /etc/neutron/neutron.conf --config-file /etc/neutron/plugin.ini
upgrade head
```
4. After this you should run: `ovs-vsctl emer-reset && service neutron-openvswitch-agent restart && service neutron-server restart` in order to apply the new configurations made on neutron openvswich agent.

After configuring this you should delete all internal and external networks and create new ones. Also all router should be recreated and configured. The source code for installing OpenStack Juno can be found at the [this link](#) .

3.3 Docker

In NUBOMEDIA, we use docker mainly for Continuous Integration (CI).

We use a Jenkins plugin named Docker plugin, that aims to provide Jenkins capability to use a docker host to dynamically provision a slave, run a single build, then tear-down that slave. We configured a Jenkins slave node that hosts all docker containers, and we

created separate jobs to do nightly build images with docker for each running environment needed in the CI system. When these jobs are done, fresh images are uploaded to Jenkins docker machine, and new slave nodes with labels are added to the Jenkins master.

The advantage of using this architecture is that Jenkins can run jobs on fresh and isolated docker containers without installing any packages or changing configurations on a live Jenkins node.

Docker can be used in NUBOMEDIA as a hypervisor driver for OpenStack, and instances can be deployed on Docker Nova Driver instead of Xen or KVM.

Docker will fetch images from OpenStack Image service (Glance), and load them onto Docker filesystem.

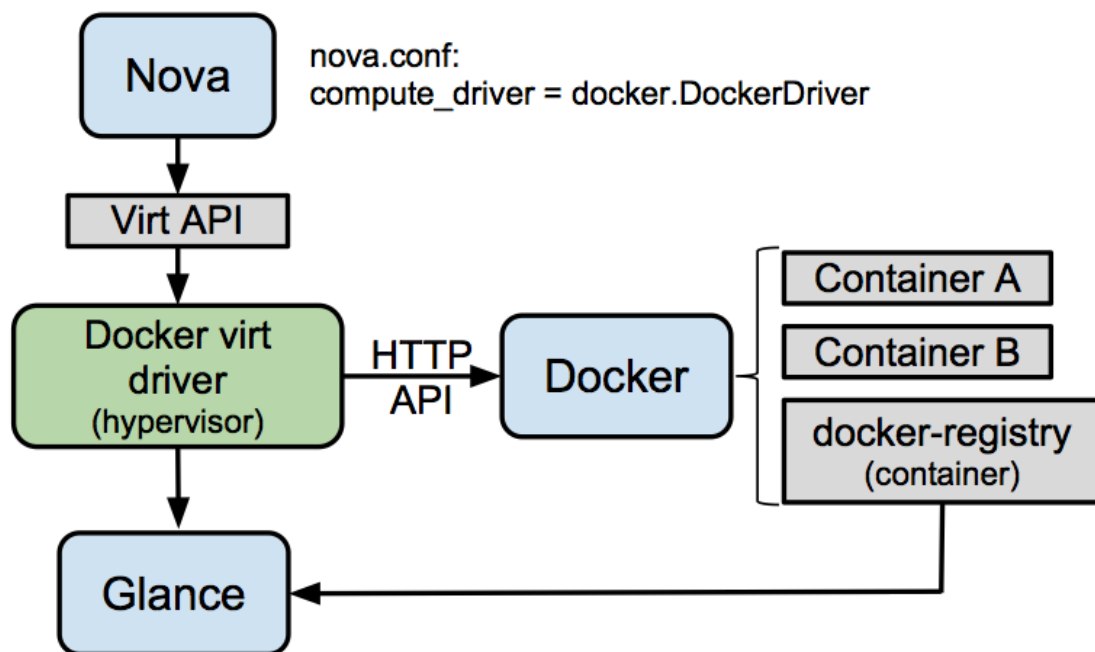


Figure 4 OpenStack with docker hypervisor architecture

Docker gives value to NUBOMEDIA project and improves the Continuous Integration process by making it much faster and flexible.

4 NUBOMEDIA API capabilities

All OpenStack projects expose an API that can be used to manage the cloud platform. Horizon (OpenStack dashboard) is also using the API to manage OpenStack projects.

4.1 Programming languages support

4.1.1 SDK

OpenStack has support on a variety of SDK in different languages. Below is a list of stable SDK

- Java: [click here to go](#)

- Node.js: [click here to go](#)
- Ruby: [click here to go](#)
- .NET: [click here to go](#)

4.1.2 REST API

OpenStack also has a REST API. In general it is recommended to use it when a SDK is not supported.

5 NUBOMEDIA API examples

This section includes a couple of examples of how to use OpenStack API. Examples are using the REST API with curl tool.

5.1 Compute API

Nova API (Compute API) is meant to launch virtual machines from images or images stored on persistent volumes.

You must first authenticate access to OpenStack services, issuing an authentication request to Keystone to acquire an authentication **token**.

Some of the more relevant API calls that can be made using the RESTful API are the following:

5.1.1 Authenticate and request token

Query:

```
curl -i 'http://80.96.122.48:5000/v2.0/tokens' -X POST -H "Content-Type: application/json" -H "Accept: application/json" -d '{"auth": {"tenantName": "nubomedia", "passwordCredentials": {"username": "nubomedia", "password": "password"}}}'
```

Result:

```
{
  "access": {
    "token": {
      "issued_at": "2014-07-15T12:37:54.907786",
      "expires": "2014-07-16T12:37:54Z",
      "id": "token_will_be_here",
      "tenant": {
        "description": "nubomedia dev",
        "enabled": true,
        "id": "fba35e226f4441c6b3b8bbd276f5d41a",
        "name": "nubomedia"
      }
    }
  },
  "serviceCatalog": [
    {
      "endpoints": [
        {
          "adminURL": "http://10.30.11.208:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a",
          "region": "RegionOne",
          "internalURL": "http://10.30.11.208:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a",
          "id": "20f2ebf7e28f45f99aa7977d8c68423d",
          "publicURL": "http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a"
        }
      ]
    },
    {
      "endpoints_links": [

```

```

    ],
    "type": "compute",
    "name": "nova"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://10.30.11.208:9696/",
        "region": "RegionOne",
        "internalURL": "http://10.30.11.208:9696/",
        "id": "15c977ac3ea142b8a082d1417ce23270",
        "publicURL": "http://80.96.122.48:9696/"
      }
    ],
    "endpoints_links": [

    ],
    "type": "network",
    "name": "neutron"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://10.30.11.208:8776/v2/fba35e226f4441c6b3b8bbd276f5d41a",
        "region": "RegionOne",
        "internalURL": "http://10.30.11.208:8776/v2/fba35e226f4441c6b3b8bbd276f5d41a",
        "id": "23adee3a4cb745649aa60884525552ab",
        "publicURL": "http://80.96.122.48:8776/v2/fba35e226f4441c6b3b8bbd276f5d41a"
      }
    ],
    "endpoints_links": [

    ],
    "type": "volumev2",
    "name": "cinderv2"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://10.30.11.208:8080",
        "region": "RegionOne",
        "internalURL": "http://10.30.11.208:8080",
        "id": "52c8d1247e0c4f06bd44582aaa2eb7bb",
        "publicURL": "http://80.96.122.48:8080"
      }
    ],
    "endpoints_links": [

    ],
    "type": "s3",
    "name": "swift_s3"
  },
  {
    "endpoints": [
      {
        "adminURL": "http://10.30.11.208:9292",
        "region": "RegionOne",
        "internalURL": "http://10.30.11.208:9292",
        "id": "1d3a4eacf15443248597686966778c1d",
        "publicURL": "http://80.96.122.48:9292"
      }
    ]
  }

```



```

    ],
    "endpoints_links":[

    ],
    "type":"image",
    "name":"glance"
  },
  {
    "endpoints":[
      {
        "adminURL":"http://10.30.11.208:8777",
        "region":"RegionOne",
        "internalURL":"http://10.30.11.208:8777",
        "id":"6dbd166306b646848e41ea05ceb3c68f",
        "publicURL":"http://80.96.122.48:8777"
      }
    ],
    "endpoints_links":[

    ],
    "type":"metering",
    "name":"ceilometer"
  },
  {
    "endpoints":[
      {
        "adminURL":"http://10.30.11.208:8776/v1/fba35e226f4441c6b3b8bbd276f5d41a",
        "region":"RegionOne",
        "internalURL":"http://10.30.11.208:8776/v1/fba35e226f4441c6b3b8bbd276f5d41a",
        "id":"8c60837c5f074b3fbce7ca26f8675217",
        "publicURL":"http://80.96.122.48:8776/v1/fba35e226f4441c6b3b8bbd276f5d41a"
      }
    ],
    "endpoints_links":[

    ],
    "type":"volume",
    "name":"cinder"
  },
  {
    "endpoints":[
      {
        "adminURL":"http://10.30.11.208:8773/services/Admin",
        "region":"RegionOne",
        "internalURL":"http://10.30.11.208:8773/services/Cloud",
        "id":"194d603e95934fbd0386cb6b5503ea9",
        "publicURL":"http://80.96.122.48:8773/services/Cloud"
      }
    ],
    "endpoints_links":[

    ],
    "type":"ec2",
    "name":"nova_ec2"
  },
  {
    "endpoints":[
      {
        "adminURL":"http://10.30.11.208:8004/v1/fba35e226f4441c6b3b8bbd276f5d41a",
        "region":"RegionOne",
        "internalURL":"http://10.30.11.208:8004/v1/fba35e226f4441c6b3b8bbd276f5d41a",

```

```

        "id": "2e971b63573d45cbbf95efb9b46bcfd6",
        "publicURL": "http://80.96.122.48:8004/v1/fba35e226f4441c6b3b8bbd276f5d41a"
    },
    "endpoints_links": [

    ],
    "type": "orchestration",
    "name": "heat"
},
{
    "endpoints": [
        {
            "adminURL": "http://10.30.11.208:8080/",
            "region": "RegionOne",
            "internalURL": "http://10.30.11.208:8080/v1/AUTH_fba35e226f4441c6b3b8bbd276f5d41a",
            "id": "621786f5605544bf96e81a73bb4380e6",
            "publicURL": "http://80.96.122.48:8080/v1/AUTH_fba35e226f4441c6b3b8bbd276f5d41a"
        }
    ],
    "endpoints_links": [

    ],
    "type": "object-store",
    "name": "swift"
},
{
    "endpoints": [
        {
            "adminURL": "http://10.30.11.208:35357/v2.0",
            "region": "RegionOne",
            "internalURL": "http://10.30.11.208:5000/v2.0",
            "id": "56c540d578d94fd8921349803eff19c7",
            "publicURL": "http://80.96.122.48:5000/v2.0"
        }
    ],
    "endpoints_links": [

    ],
    "type": "identity",
    "name": "keystone"
}
],
"user": {
    "username": "nubomedia",
    "roles_links": [

    ],
    "id": "ac4b3be6d79b4fb4b1d25361ed1c75b9",
    "roles": [
        {
            "name": "_member_"
        }
    ],
    "name": "nubomedia"
},
"metadata": {
    "is_admin": 0,
    "roles": [
        "9fe2ff9ee4384b1894a90878d3e92bab"
    ]
}

```

```
}
}
}
```

5.1.2 Lists IDs, names, and links for all servers

When you have a list with all API endpoints and the authentication ticket, you can query the Nova API to get a list of all servers.

Query:

```
curl -v -H "X-Auth-Token:token" http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers
```

Result:

```
{
  "servers": [
    {
      "id": "d6acf269-073d-4a0c-ad52-d3655fd601b8",
      "links": [
        {
          "href": "http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8",
          "rel": "self"
        },
        {
          "href": "http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8",
          "rel": "bookmark"
        }
      ],
      "name": "Alin"
    }
  ]
}
```

5.1.3 Gets details for a specified server

Query:

```
curl -v -H "X-Auth-Token:token" http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8
```

Result:

```
{
  "server": {
    "status": "ACTIVE",
    "updated": "2014-07-11T20:05:29Z",
    "hostId": "e70129f52231eab4ddeb0106b08996e6f0e35f3b8819cdc4a2278bfe",
    "addresses": {
      "internal_nubomedia": [
        {
          "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:94:62:49",
          "version": 4,
          "addr": "172.22.2.18",
          "OS-EXT-IPS:type": "fixed"
        },
        {
          "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:94:62:49",
          "version": 4,

```

```

        "addr": "80.96.122.53",
        "OS-EXT-IPS:type": "floating"
    }
]
},
"links": [
    {
        "href": "http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8",
        "rel": "self"
    },
    {
        "href": "http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8",
        "rel": "bookmark"
    }
],
"key_name": "Alin",
"image": {
    "id": "e56d9276-c0f0-4d2b-a466-065bc8b94551",
    "links": [
        {
            "href": "http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/images/e56d9276-c0f0-4d2b-a466-065bc8b94551",
            "rel": "bookmark"
        }
    ]
},
"OS-EXT-STS:task_state": null,
"OS-EXT-STS:vm_state": "active",
"OS-SRV-USG:launched_at": "2014-07-11T20:05:29.0* Connection #0 to host 80.96.122.48 left intact
00000",
"flavor": {
    "id": "75077619-d2c0-4e6a-83e9-e6ea52d62a36",
    "links": [
        {
            "href": "http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/flavors/75077619-d2c0-4e6a-83e9-e6ea52d62a36",
            "rel": "bookmark"
        }
    ]
},
"id": "d6acf269-073d-4a0c-ad52-d3655fd601b8",
"security_groups": [
    {
        "name": "default"
    }
],
"OS-SRV-USG:terminated_at": null,
"OS-EXT-AZ:availability_zone": "nova",
"user_id": "0da3590746e744b494c72914c6ed2870",
"name": "Alin",
"created": "2014-07-11T20:05:16Z",
"tenant_id": "fba35e226f4441c6b3b8bbd276f5d41a",
"OS-DCF:diskConfig": "AUTO",
"os-extended-volumes:volumes_attached": [

],
"accessIPv4": "",
"accessIPv6": "",
"progress": 0,

```

```
"OS-EXT-STS:power_state":1,
"config_drive":"","
"metadata":{
}
}
}
```

5.1.4 Lists networks and addresses for a specified tenant and server

Query:

```
curl -v -H "X-Auth-Token:token"
http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8/ips
```

Result:

```
{
  "addresses":{
    "internal_nubomedia":[
      {
        "version":4,
        "addr":"172.22.2.18"
      },
      {
        "version":4,
        "addr":"80.96.122.53"
      }
    ]
  }
}
```

5.1.5 Hard reboot instance

Query:

```
curl -v POST -d '{"reboot": {"type": "HARD"}}' --header "Content-Type:application/json" -H "X-Auth-Token:token"
http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers/d6acf269-073d-4a0c-ad52-d3655fd601b8/action
```

Result:

The instance with id d6acf269-073d-4a0c-ad52-d3655fd601b8 will be hard rebooted.

5.1.6 Lists all details for available flavors

```
curl -v -H "X-Auth-Token:token" http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/flavors/detail
```

Query:

Result:

```
{
  "flavors":[
    {
      "name":"m1.tiny",
      "links":{
```

```

    {
      "href":"http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/flavors/1",
      "rel":"self"
    },
    {
      "href":"http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/flavors/1",
      "rel":"bookmark"
    }
  ],
  "ram":512,
  "OS-FLV-DISABLED:disabled":false,
  "vcpus":1,
  "swap": "",
  "os-flavor-access:is_public":true,
  "rxtx_factor":1.0,
  "OS-FLV-EXT-DATA:ephemeral":0,
  "disk":1,
  "id":"1"
},
{
  "name":"m1.small",
  "links":[
    {
      "href":"http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/flavors/61354281-03bd-4ac4-8e1a-f9bc13795a09",
      "rel":"self"
    },
    {
      "href":"http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/flavors/61354281-03bd-4ac4-8e1a-f9bc13795a09",
      "rel":"bookmark"
    }
  ],
  "ram":2048,
  "OS-FLV-DISABLED:disabled":false,
  "vcpus":1,
  "swap": "",
  "os-flavor-access:is_public":true,
  "rxtx_factor":1.0,
  "OS-FLV-EXT-DATA:ephemeral":0,
  "disk":6,
  "id":"61354281-03bd-4ac4-8e1a-f9bc13795a09"
},
{
  "name":"m1.medium",
  "links":[
    {
      "href":"http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/flavors/75077619-d2c0-4e6a-83e9-e6ea52d62a36",
      "rel":"self"
    },
    {
      "href":"http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/flavors/75077619-d2c0-4e6a-83e9-e6ea52d62a36",
      "rel":"bookmark"
    }
  ],
  "ram":4096,
  "OS-FLV-DISABLED:disabled":false,
  "vcpus":2,
  "swap": "",

```

```

      "os-flavor-access:is_public":true,
      "rxtx_factor":1.0,
      "OS-FLV-EXT-DATA:ephemeral":0,
      "disk":10,
      "id":"75077619-d2c0-4e6a-83e9-e6ea52d62a36"
    },
    {
      "name":"m1.xlarge",
      "links":[
        {
          "href":"http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/flavors/bccf44c9-38e9-4771-b7f0-8a8776c89765",
          "rel":"self"
        },
        {
          "href":"http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/flavors/bccf44c9-38e9-4771-b7f0-8a8776c89765",
          "rel":"bookmark"
        }
      ],
      "ram":16384,
      "OS-FLV-DISABLED:disabled":false,
      "vcpus":8,
      "swap":"",
      "os-flavor-access:is_public":true,
      "rxtx_factor":1.0,
      "OS-FLV-EXT-DATA:ephemeral":0,
      "disk":20,
      "id":"bccf44c9-38e9-4771-b7f0-8a8776c89765"
    },
    {
      "name":"m1.large",
      "links":[
        {
          "href":"http://80.96.122.48:* Connection #0 to host 80.96.122.48 left intact
8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/flavors/cc8f8013-5ba6-4ab2-b362-c41b1506d2d3",
          "rel":"self"
        },
        {
          "href":"http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/flavors/cc8f8013-5ba6-4ab2-b362-c41b1506d2d3",
          "rel":"bookmark"
        }
      ],
      "ram":8192,
      "OS-FLV-DISABLED:disabled":false,
      "vcpus":4,
      "swap":"",
      "os-flavor-access:is_public":true,
      "rxtx_factor":1.0,
      "OS-FLV-EXT-DATA:ephemeral":0,
      "disk":15,
      "id":"cc8f8013-5ba6-4ab2-b362-c41b1506d2d3"
    }
  ]
}

```

5.1.7 Create Instance

Query:

```
curl -i http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers \
-X POST \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-H "X-Auth-Token:token" \
-d '{"server": { "name": "nubomedia-test", "imageRef": "e56d9276-c0f0-4d2b-a466-065bc8b94551", "flavorRef":
"75077619-d2c0-4e6a-83e9-e6ea52d62a36", "max_count": 1, "min_count": 1, "key_name": "Alin",
"security_groups": [ { "name": "AllPortsOpen" } ], "networks": [{"uuid": "fd704f1b-9238-4c2c-a0f5-4ffb4543e33a"}]
}}'
```

Result:

```
{
  "server":{
    "security_groups":[
      {
        "name":"AllPortsOpen"
      }
    ],
    "OS-DCF:diskConfig":"MANUAL",
    "id":"433ac0e9-75bb-40ad-9822-21c71abaa3e7",
    "links":[
      {
        "href":"http://80.96.122.48:8774/v2/fba35e226f4441c6b3b8bbd276f5d41a/servers/433ac0e9-75bb-40ad-
9822-21c71abaa3e7",
        "rel":"self"
      },
      {
        "href":"http://80.96.122.48:8774/fba35e226f4441c6b3b8bbd276f5d41a/servers/433ac0e9-75bb-40ad-
9822-21c71abaa3e7",
        "rel":"bookmark"
      }
    ],
    "adminPass":"9im3yAvVnm8r"
  }
}
```

5.2 Neutron API

Neutron REST API provides HTTP service that uses all aspects of the HTTP protocol including methods, URIs, media types, response codes, and so on. Clients can use all existing features of the protocol including caching, persistent connections, and content compression. For example, providers who employ a caching layer can respond with a 203 code instead of a 200 code when a request is served from the cache. Additionally, clients can offer support for conditional GET requests by using ETags, or they may send a redirect in response to a GET request.

5.2.1 Authentication and authorization

Neutron API uses the Keystone Identity Service as the default authentication service. When Keystone is enabled, users that submit requests to the OpenStack Neutron service must provide an authentication token in X-Auth-Token request header. You obtain the token by authenticating to the Keystone endpoint. For more information about Keystone, see the chapter [4.1.1 Authenticate and request token](#).

When Keystone is enabled, the `tenant_id` attribute is not required in create requests because the tenant ID is derived from the authentication token.

The default authorization settings allow only administrative users to create resources on behalf of a different tenant.

Neutron uses information received from Keystone to authorize user requests and handles the following types of authorization policies:

- Operation-based policies specify access criteria for specific operations, possibly with fine-grained control over specific attributes.
- Resource-based policies access a specific resource. Permissions might or might not be granted depending on the permissions configured for the resource. Currently available for only the network resource.

The actual authorization policies enforced in OpenStack Neutron might vary from deployment to deployment.

5.2.2 Filtering the response and column selection

The Neutron API supports filtering based on all top level attributes of a resource. Filters are applicable to all list requests. For example the following query will return all the networks whose name is internal. When multiple filters are specified, the Neutron API will return only objects that satisfy all the filters, this applying an AND condition among filters. Neutron does not offer a mechanisms for ORing filters. To this aim, the user can submit a distinct request for each filters, and then build a set on the client-side from received responses.

Query:

```
curl -v -H "X-Auth-Token:token" http://80.96.122.48:9696/v2.0/networks?name=internal
```

Result:

```
{
  "networks": [
    {
      "status": "ACTIVE",
      "subnets": [
        "e30a288e-af8f-49db-9540-072a8de7de4e"
      ],
      "name": "internal",
      "router:external": true,
      "tenant_id": "9e9997a5432143c79a9d0f4f6c56173f",
      "admin_state_up": true,
      "shared": false,
      "id": "12241746-5465-45fa-ac39-bd1fdbdb9b0e"
    }
  ]
}
```

By default, Neutron returns all attributes for any Show or List call. The Neutron API has a mechanism to limit the set of attributes returned (e.g., return just 'id'). Attributes returned from the Neutron API can be controller using the fields query parameter. For example the following query:

Query:

```
curl -v -H "X-Auth-Token:token" http://80.96.122.48:9696/v2.0/networks?fields=name
```

Result:

```
{
  "networks":[
    {
      "name":"internal"
    },
    {
      "name":"external"
    },
    {
      "name":"internal_nubomedia"
    }
  ]
}
```

5.2.3 Neutron asynchronous behavior

Neutron API presents a logical model of network connectivity consisting of networks, ports, and subnets. It is up to the Neutron plugin to communicate with the underlying infrastructure to ensure packet forwarding is consistent with the logical model and it might perform these operations asynchronously. This means that when an API client modifies the logical model using an HTTP POST, PUT, or DELETE, the API call may return prior to the plugin performing any modifications to underlying virtual and/or physical switching devices.

Example:

Let's consider the case where a client uses an HTTP PUT to set the attachment for a port. There is no guarantee that packets sent by the interface named in the attachment will be forwarded immediately once the HTTP call returns. However, there is a guarantee that a subsequent HTTP GET to view the attachment on that port would return the new attachment value. The "status" attribute, available for network and port resources might be used to understand whether the Neutron plugin has successfully completed the configuration of the interested resource.

5.3 Telemetry API

With this API can be managed telemetry operations.

Current version is v2:

<http://developer.openstack.org/api-ref-telemetry-v2.html>

<http://docs.openstack.org/developer/ceilometer/webapi/v2.html>

5.3.1 Meters

5.3.1.1 *Get list of meters*

Full list here: <http://docs.openstack.org/developer/ceilometer/measurements.html>

GET /v2/meters

Query:

```
curl -i -X GET http://80.96.122.48:8777/v2/meters -H "User-Agent: python-keystoneclient" -H "X-Auth-Token: token"
```

Result:

```
{
  "user_id": "de4e087a406a429689e216b412b4e911",
  "name": "network",
  "resource_id": "c31bb43c-85bd-4252-9c0a-ce1218412563",
  "source": "openstack",

  "meter_id": "YzMxYmIOM2MtODViZC00MjUyLTljMGEtY2UxMjE4NDYNTYzK25ldHdvcmcs=\n",
  "project_id": "e441c2bb64d14f1a8af49c2fde06998f",
  "type": "gauge",
  "unit": "network"
},
{
  "user_id": "de4e087a406a429689e216b412b4e911",
  "name": "network.create",
  "resource_id": "c31bb43c-85bd-4252-9c0a-ce1218412563",
  "source": "openstack",

  "meter_id": "YzMxYmIOM2MtODViZC00MjUyLTljMGEtY2UxMjE4NDYNTYzK25ldHdvcmcsuY3JlYXRl\n",
  "project_id": "e441c2bb64d14f1a8af49c2fde06998f",
  "type": "delta",
  "unit": "network"
},
```

5.3.1.2 *Get samples of metric for a specific instance*

```
GET /v2/meters/{meter_name}
Instance id is: cf67f3c6-c840-45e3-81e6-5cbfab443f68
```

The following query will get all samples for cpu utilization of specified instance after 2014-07-15T13:34:17

Query:

```
curl -X GET "http://80.96.122.48:8777/v2/meters/cpu_util?q.field=resource_id&q.op=eq&q.value=cf67f3c6-c840-45e3-81e6-5cbfab443f68&q.field=timestamp&q.op=gt&q.value=2014-07-15T13:34:17" -H "X-Auth-Token: TOKEN"
```

Result:

```
{
  "counter_name": "cpu_util",
  "user_id": "0da3590746e744b494c72914c6ed2870",
  "resource_id": "cf67f3c6-c840-45e3-81e6-5cbfab443f68",
  "timestamp": "2014-07-13T07:14:56",
  "recorded_at": "2014-07-13T07:14:56.956000",
  "resource_metadata": {},
  "source": "openstack",
  "counter_unit": "%",
  "counter_volume": 11.6275,
  "project_id": "9e9997a5432143c79a9d0f4f6c56173f",
  "message_id": "64ed6a5a-0a5d-11e4-9ccc-001a648fa98a",
  "counter_type": "gauge"
}, {
  "counter_name": "cpu_util",
  "user_id": "0da3590746e744b494c72914c6ed2870",
  "resource_id": "cf67f3c6-c840-45e3-81e6-5cbfab443f68",
  "timestamp": "2014-07-13T07:04:56",
  "recorded_at": "2014-07-13T07:04:56.836000",
  "resource_metadata": {},
  "source": "openstack",
  "counter_unit": "%",
  "counter_volume": 58.746666666666663,
  "project_id": "9e9997a5432143c79a9d0f4f6c56173f",
  "message_id": "ff3a5ba6-0a5b-11e4-9ccc-001a648fa98a",
  "counter_type": "gauge"
},
```

6 References

- [1] Openstack: Open source software for creating public and private clouds. See <http://www.openstack.org/>.
- [2] OpenNebula platform: <http://opennebula.org>
- [3] Apache CloudStack <http://cloudstack.apache.org>
- [4] Docker <https://www.docker.com>
- [5] CoreOS <https://coreos.com>
- [6] Project Atomic from RedHat <http://www.projectatomic.io>
- [7] Kubernetes - Linux containers manager <http://kubernetes.io>
- [8] RedHat OpenStack – <http://openstack.redhat.com/Quickstart>