

NETWORK FUNCTION VIRTUALIZATION AS A PART OF MODULAR VIRTUALIZED INFRASTRUCTURE IN LAB ENVIRONMENT

V. Šac¹, L. Havaš¹, D. Srpak¹, E. Tomičić²

¹University North (CROATIA)

²Graditeljska škola Čakovec (CROATIA)

Abstract

Universities research environment often need flexible IT infrastructure for network services research as well as isolated lab working environments for students. Different research environment groups and applications have to be moved in cloud to be able to allocate resource on demand, e.g. number of CPU cores. Also, a system that supports flexible and automated environment is needed, which can be installed / reinstalled in couple of minutes supporting computing nodes and upgraded as research activities grows with no service discontinuity. As Lab virtualized infrastructure, OpenStack as common open source platform is used, built on UNIX based operating system and Intel based hardware.

This paper analyses known industry approaches and virtualization techniques resulting with design of the lab virtualized infrastructure that are going to enable flexible research working environment. Network function virtualization as key part of such modular environment should improve performance and security, as well as change physical network components.

The usage of Virtual Machines (VM) or containers are key elements inside Lab infrastructure in order to provide an abstraction between hardware resources and applications. Before LAB design creation as the final result, basics items and steps have to be adopted and defined, such as standardisation of students/research groups that can interact with the system, self-managing and orchestration. Common building blocks are used as parts of design such as hardware based on x86 industry, hypervisors, UNIX based operating system, databases, high performance data model.

The main goal is design of a lab environment, which enable flexible, secure, high available and self-growing platform where different research groups and students can develop and test their systems and get reliable results.

Keywords: Network Function Virtualization, Virtualized Infrastructure, multi tenancy, reliability.

1 INTRODUCTION

It is well known that lab environment at faculties and universities needs flexible IT infrastructure for network services research inside different research branches as for example Edge Services (vPE, vCPE), Mobile (IMS,5G), Internet of Things (IoT).

By disconnecting physical aspects of hardware from its functionality, we significantly increase the domain of its operation and allow installing different operating systems on the same server. Installing such servers is quick and relatively easy and allows easy migration to other physical locations.

One of the prerequisites for realizing network function virtualization within university labs is the implementation of cloud infrastructure. All applications should be moved in cloud to be able to allocate resource on demand (number of CPU cores e.g.). Network functions needs monitoring, availability and avoid single point of failures [1]-[3].

System that supports flexible and automated environment, which also can be installed/reinstalled in couple of minutes is needed.

Open source solutions will be used in the lab design. Infrastructure as a service (IaaS) such as OpenStack [4], [5] built on UNIX based operating system and Intel based HW makes a main platform for building research environment including network function virtualization [6]-[9].

Before LAB NFVI (Network Functions Virtualization Infrastructure) design it is necessary to adopt and define basics items and steps:

- Standardisation of users/research groups that can interact with the system through a user-friendly web *interface*, a command line interface or a software API. Tenancy [4] and security configuration inside OpenStack should be prepared according internal plans and needs.
- Fine-grained control of the extent to which physical resources are shared. Flexibility and resource dynamic sharing (CPU, memory, file system) [10] must be included.
- Self-managing and orchestration; Research groups organized in tenants as isolated security environments can self-serve and boot a software image without requiring operator assistance.
- Additional security as user has a higher degree of separation from each other Members/tenants of different projects needs different security configuration. Projects are separated one from another and may need different networks.
- Design of an HPC (High Performance Computing) based on OpenStack will enable minimal resource overhead and will give possibility to dynamically reorganize tenant/projects as system will grow in time. Way of virtualisation keeps balance between new demands for resources and consequential overhead.

The basic blocks, from which the modular virtualized infrastructure of multifunctional lab is built, are shown in Fig. 1.

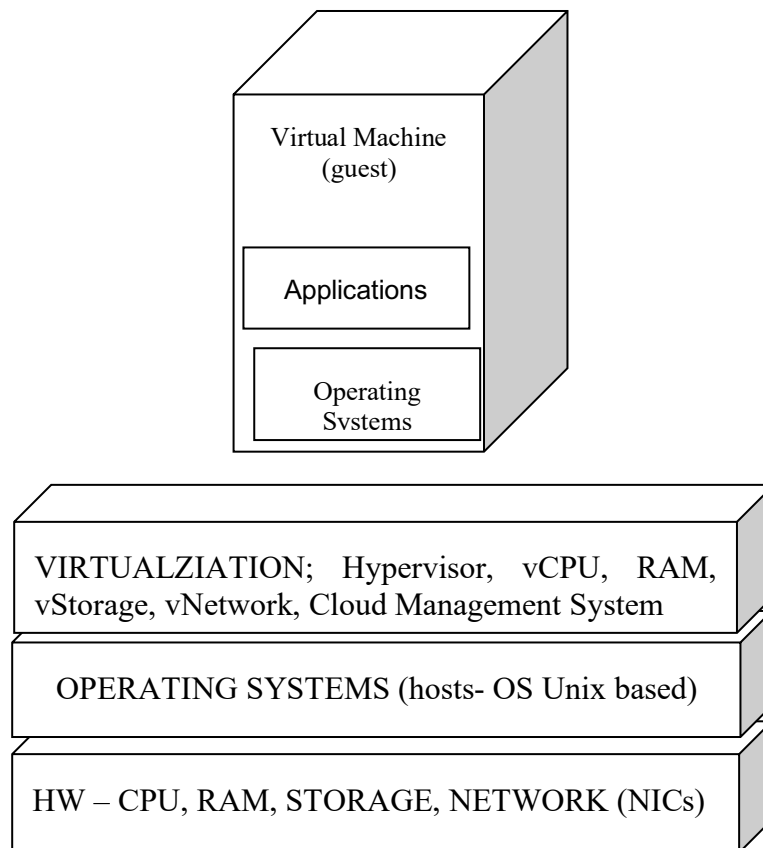


Figure 1. Basic Lab building blocks - from HW to VM

2 LAB USER AND GROUP STANDARDIZATION REQUIREMENTS

From internal (university) organization we have ongoing request for dynamical IT environment changes. Let us make two basics standardized group of users in our IT infrastructure: students and researches. Students need educational environments where they can develop, run and test different tasks from different areas and courses (IP networks, network emulations, network protocol analyses, program developing, coding etc.). They also need separated working environments from each other, which require security configuration, personal credentials and personal CPU, RAM and disk space resources. This is solved in today's common lab by assigning personal computer to each student which means only one operating system is available. Second group are researchers (professors and students) working on dedicated projects but they are also equipped with personal computers

depending on present configuration which cannot be dynamically changed. For example, some DB research and coding could be done on operating system A (non- Unix) which is available in Lab but the same network research team need Unix like OS to run some network security tests (Firewall testing for instance). Such requirements tend to increase budget on daily basis therefore unique infrastructure is needed which support multi OS environment. OpenStack as open source platform enables learning, development and research in same time. Defined users and groups as tenants are placed inside OpenStack Project (previous versions called tenants).

3 INFRASTRUCTURE PLATFORM FOR RUNNING LAB

In very basic approach we need to have minimum three types of networks. Management network, that connects management ports (Management network- using Intelligent Platform Management Interface ports, IPMI) which are necessary for initial HW installation, configuration, management and basic HW monitoring.

OpenStack management network (Fig.2) enables installation, monitoring and management for operating systems and OpenStack modules (Keystone, Glance, Nova, Cinder, Neutron [4], [5].)

VM as basic product for running network application and other services will use switches on the top of the rack with higher bandwidth to enable maximum application performance (Fig2.). Application networks are connecting applications that are running inside virtual machines.

Following the Network Function Virtualization platforms [11]-[13] three basic separations of HW resources are defined as Compute, Controlling and Storage nodes. There is certain flexibility when installing different type of virtual blocks on particular computer. In general for Lab purposes we separate Compute nodes, in the terms of OpenStack, should be installed on HW with high CPU and memory resources. Compute node disk space is enough just to run operating systems and additional compute node SW.

Storage nodes should be equipped with more disk space because of storing different VM images which requires more disk space depends on sizing limits.

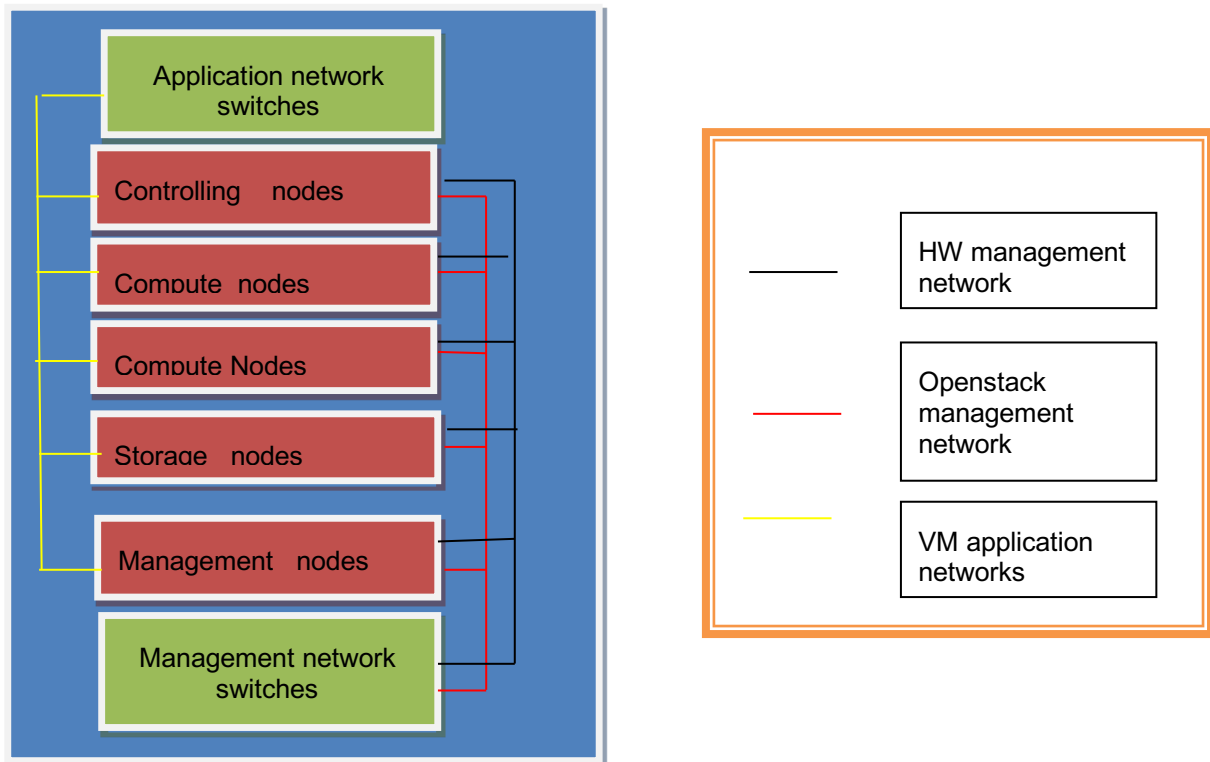


Figure 2. Lab design: Inside one rack with possible future scalability.

Controller nodes [4] which are running the most of OpenStack shared services, (nova scheduler for instance), could be placed and combined (inside same HW) together with Compute nodes in the case of small lab environments, but when including performance requirements, separation of the Compute and Controlling functionalities into the separate HW nodes- servers is highly recommended.

VMs running on OpenStack infrastructure will use different networks and will running different network applications, switches, and firewalls. Neutron [5], [10] as main network configuration interface module is used to create separate networks. Neutron [5], [10] as part of OpenStack will keep data for each created network and subnet inside relational database. OpenStack relational database as core part of infrastructure including other modules should be installed in high available mode which needs replication of modules across at least three physical nodes [10] which will prevent lab infrastructure in case of failure of single physical server.

Bare metal commodity servers should be based on x86 industry and configuration (number of CPUs, RAM, Disk space) should follow lab optimal sizing, functionality, and needed performance. Fig. 2 shows basic HW building blocks which need to be scalable in case of future infrastructure extension need.

Sizing for VMs are part of resource grouping module (number of vCPUs, RAM and disk space) inside OpenStack flavor [4]. Each flavor will define compute resource for running instance.

By summarizing proposed core solution shown in Fig 2. Lab has three types of nodes:

- 1 Compute nodes, high CPU speeds and RAM, disk space needed only for OS
- 2 Storage nodes, large disk space, data storing functionality
- 3 Controller nodes running cloud control management SW

Network module that are running open source network planes (open switch, and Neutron for instance) should be integrated within existing HW nodes with possibility to be moved on separate HW node, if needed.

Module like Ceph [14] will be used for high performance data model. VMs will use high performance protocols to enable maximum network performance (SR-IOV -Single Root Input Output Virtualization protocol e.g. [15])

Host operating system running inside virtual machines is 64 bit Unix based operating system. KVM [4] is chosen as hypervisor, Guest operating system recommendation for research groups is any open source Linux, Ubuntu, CentOS, FreeBSD. OpenStack controls compute node pools where guest VMs are running. Relational database will keep core of OpenStack data system [5].

4 MANAGE LAB ENVIRONMENT

To enable independent Lab management space we need separate HW-Management nodes (Fig 3.) Management host uses only KVM as hypervisor to run 2 basic VMs. One VM is used as Lab access node with connection to the public network. Another management VM is used to enable boot process for OpenStack which will use automation scripts and development tools to run all services in very short time. Management VM also have OpenStack client SW installed. Insisting on lab automation and orchestration from very beginning will enable flexible management in future. Most frequently request for services running in lab environment is to change modules, to do the upgrade, downgrade for the purpose of testing and researching. Working on automation development is also part of student educational program.

In order to build infrastructure we have summarized above and need to improve some performance testing measuring.

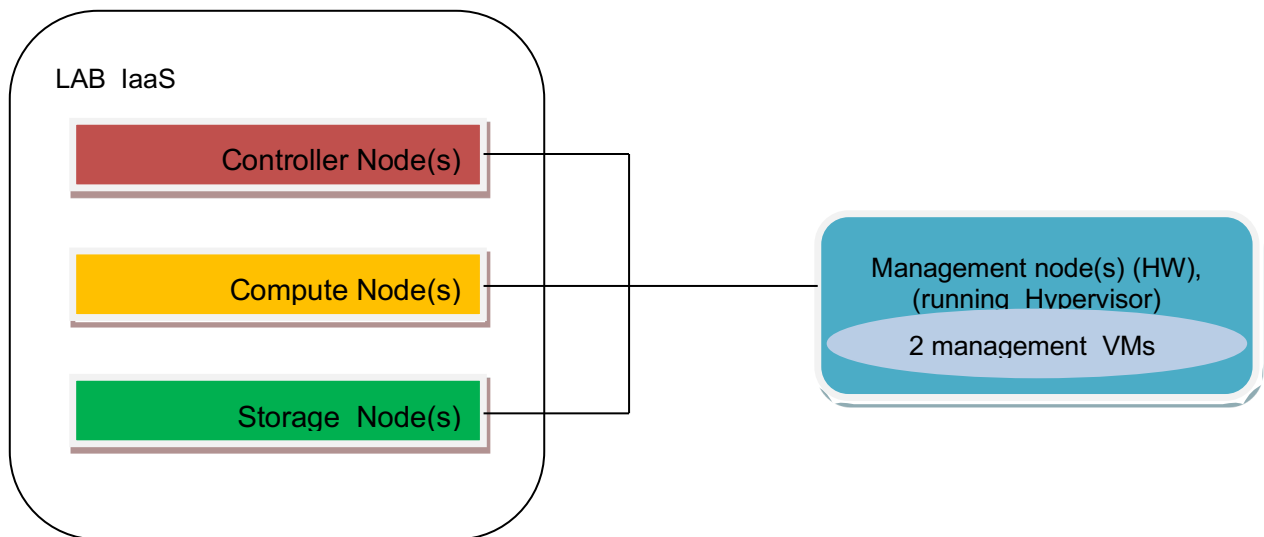


Figure 3. Installing, configuring and managing Lab from separate management nodes.

5 INSTALLATION DEVELOPMENT EVOLVES FROM MANUAL TO AUTOMATED PROCESS

Manual installation procedure as follows will be changed later with higher degree of automation as part of development process. We can highlight several phases of installation:

- Installing HW, connect all components.
- Installing Operation Systems (Bare metal servers can use pxe boot, different tools and scripts for automation installation process.
- Install OpenStack modules, network and hypervisor layer respectively to HW node function.
- Define tenants/projects [10], users, roles and resources per project (number of CPUs, memory, disk space)
- Define and create flavour and networks (by using Neutron module [10])
- Apply Heat template/roles over cloud definition
- Create/Import VNF images, spin up VM instances (use automation templates , Heat [4])
- Installing and testing network services – NFV inside virtual machines

5.1 Tenants/projects and quotas inside IaaS platform

OpenStack [4] supports multiple tenants/projects; a concept can be applied to have different sets of services running in Lab. Each set of network services and applications as part of a particular research group is running inside a project on OpenStack level, and each tenant has defined limits and quotas. Orchestrator should be able to take care of monitoring tenant quota usage and refine them based on different research group demands. Using security configuration inside OpenStack Identity service (Keystone [4]) we can isolate data sharing to a particular project.

Orchestrations at the beginning of lab deployment process should enable VM create, start, stop, and delete process.

Orchestration has to store local data about current state of Lab such as:

- running virtual machines
- service state
- logs (security and troubleshooting)
- performance history and resource usage
- basic service configuration

State of the services and virtual machines in lab may change in the case of orchestrator failure, so it is necessary to rebuild the knowledge about running VMs and services upon start of the orchestrator. Current state may be compared to the desired state which is stored permanently and corrective actions may be taken as they would be in normal circumstances. Logs collection on central place is mandatory for troubleshooting and to perform analysis. Developing orchestration in lab should be part of designated project which will cover process of automation on several levels by using development tools.

The usage of Virtual Machines (VM) is a key element inside Lab infrastructure in order to provide an abstraction between hardware resources and applications.

Network services/applications testing and development are running inside VMs.

Orchestration shall be based on open source software components that are built or integrate inside the OpenStack. Orchestration includes standardized automation procedures, scripts for provisioning (as we mentioned OpenStack Heat project [4])

From HW prospective, main building blocks Fig. 2 (one rack model) should be connected to physical routers, and switches divided in two network plains: management and application networks. Management network have secure separation (firewall) from application traffic (switches on the top of the rack) and enables managing physical servers and also virtual machines. Application switches have high network bandwidth (10Gbps and more) connecting compute nodes, and storage nodes.

6 AUTOMATION STEPS FOR LAB BUILDING PROCESS

Lab (re)building is a task which has to be automated since it may help avoiding human errors and drastically decrease time and resources required to get the lab environment functional. In short, building automation can be decomposed into the following steps:

- 1 Make hardware and physical installation.
- 2 Obtaining data necessary to configure initial boot (MAC addresses, router ports, IP addresses attached to module name, written in configuration files).
- 3 Uploading the data about HW into the boot server.
- 4 Starting an initial boot, operating system and OpenStack installation, starting their configuration (running scripts).
- 5 Verification of installation success by starting services, checking progress and measuring performance (automated check tools are also available [4])

Lab may be rebuilt when necessary; scripts and templates can be written for different installation cases. Steps 1-3 can be replaced with shutdown of an existing system after which reinstallation can be run with new configuration files.

7 RESULTS OF GIVEN LAB MODEL

Proposed lab model based on open source IaaS (Infrastructure as a Service) enables organizing and structuring lab work according different research group needs.

Benefits that we have from lab model proposed above are:

- Support for research groups is achieved and flexible
- Central point for data and application share is achieved and optimized
- Network functions virtualization can be easy loaded inside VM
- Different educational groups organized in tenants/projects are able to support different courses and laboratory excises in secure and isolated way
- Working environment can be reinstalled in short period of time and enables running different applications inside different operating systems inside VMs (Fig. 4.).

```

root@OSVZ-1: ~/scripts
heat template version: 2017-09-01
Description: Launch a basic instance with qcow2 net_emulation image using the
            ``ml.middle`` flavor, ``mykey`` key, and one network.

parameters:
  NetID:
    type: string
    description: Network ID to use for the instance.

resources:
  server:
    type: OS::Nova::Server
    properties:
      image: net_emulation
      flavor: ml.middle
      key_name: mykey
      networks:
        - network: ( get_param NetID )

outputs:
  instance_name:
    description: Name of the instance.
    value: ( get_attr: [ server, name ] )
  instance_ip:
    description: IP address of the instance.
    value: ( get_attr: [ server, first_address ] )

"heat-lab-sample.yaml" 27L, 686C

```

```

root@OSVZ-1: ~/scripts
root@OSVZ-1:~/scripts# clear
root@OSVZ-1:~/scripts# openstack stack create -t heat-lab-sample.yaml --parameter "NetID=b5b72b5b-5c4f-44ed-aa49-660097f2a643" net_emulation_1
+-----+-----+-----+-----+-----+-----+-----+
| Field | Value |
+-----+-----+-----+-----+-----+-----+
| id | 28c6e7e2-c67d-44ab-8324-4f6f91477b5b |
| stack_name | net_emulation_1 |
| description | Launch a basic instance with qcow2 immunes image using the ``ml.nano`` flavor, ``mykey`` key, and one network. |
| creation_time | 2019-05-06T15:32:19Z |
| updated_time | None |
| stack_status | CREATE_IN_PROGRESS |
| stack_status_reason | Stack CREATE started |
+-----+-----+-----+-----+-----+-----+

root@OSVZ-1:~/scripts# openstack stack list
+-----+-----+-----+-----+-----+-----+
| ID | Stack Name | Project | Stack Status | Creation Time | Updated Time |
+-----+-----+-----+-----+-----+-----+
| 28c6e7e2-c67d-44ab-8324-4f6f91477b5b | net_emulation_1 | 96176b0446824d46b7925334d50e97c7 | CREATE_IN_PROGRESS | 2019-05-06T15:32:19Z | None |
| 843odd38-836d-4f76-8e90-2249b461017E | net_emulation | 96176b0446824d46b7925334d50e97c7 | CREATE_COMPLETE | 2019-05-06T15:30:02Z | None |
| 32885d06-4494-44a4-b06b-cfe13c2dbdcf | test-stack | 96176b0446824d46b7925334d50e97c7 | CREATE_COMPLETE | 2019-04-26T10:25:56Z | None |
| e13645df-46bc-432a-a0c9-7854c9be912e | stack1 | 96176b0446824d46b7925334d50e97c7 | CREATE_COMPLETE | 2019-04-26T17:32:25Z | None |
| a9eb418b-d6a5-4009-95b2-feddd0903ad8 | stack | 96176b0446824d46b7925334d50e97c7 | CREATE_COMPLETE | 2019-04-24T12:41:59Z | None |
+-----+-----+-----+-----+-----+-----+

```

Figure 4. An example from lab model: VM creation by using Heat orchestration template [4].

The main achievement with proposed model is an environment which enable flexible, secure, high available and self growing infrastructure where different research groups can develop and test their systems and get reliable results.

Problems that we noticed so far are complexity in update and upgrade process, but it should be part of future work.

8 CONCLUSION

As first step in reorganizing our lab we proposed acceptable designed model. It is opening opportunity of learning and developing in the area of VNF infrastructure installation, automation and network service orchestration once infrastructure is up and running. Such a model enables also education and research out of vendor boxes and propriety software solutions.

NFV implements network functions through software virtualization techniques and runs them on commodity hardware (i.e., industry standard servers, storage, and switches)” [11]

Proposed lab design enables dynamic and orchestrated VM provisioning. Testing of different virtualized network function, such as vCPE , or vSwitch could be done by simple image copy and deploy inside lab [16]. Orchestration on VM level could be done by using scripting or development tools or by acquiring templates and open code from different projects.

HW appliances from different vendors, costs, electrical power consumption and vendor dependencies does not give us flexibility and stands as main barrier for future lab development, research and education. In the same time these are main drivers for changing existing infrastructure. Making

design of infrastructure that is based on open source architecture, automation during installation and configuration process will enable new possibilities for emerging different technologies such as network virtualization.

REFERENCES

- [1] B. Yi, X. Wang, K. Li, S. k.Das, M. Huang, "A comprehensive survey of Network Function Virtualization," *Computer Networks*, vol. 133, pp. 212-242, 2018.
- [2] D. Hall, "Ansible Configuration Management", Second Edition, Packt Publishing Ltd., Birmingham, April 2015.
- [3] T. A. Limoncelli, S. R. Chalup, C. J. Hogan, "The Practice of Cloud System Administration, Volume 2", Addison-Wesley Professional, Indiana, 2014.
- [4] Openstack, <https://www.openstack.org>, Accessed on 01.04.2019.
- [5] J. Denton, "Learning OpenStack Networking (Neutron)," Second edition, Packt Publishing Ltd., Birmingham, 2015.
- [6] W. Rankothge, J. Ma, F. Le, A. Russo, J. Lobo, "Towards Making Network Function Virtualization a Cloud Computing Service," *IFIP/IEEE International Symposium on Integrated Network Management (IM2015)*, Ottawa, Canada, pp. 89-97, May 2015.
- [7] Y. Li, M. Chen, "Software-Defined Network Function Virtualization: A Survey", *IEEE Access*, vol. 3, pp. 2542-2553, 2015.
- [8] N. M. K. Chowdhury, R. Boutaba, "Network Virtualization: State of the Art and Research Challenges", *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20-26, 2009.
- [9] T. Wood, K. K. Ramakrishnan, J. Hwang, G. Liu, W. Zhang "Toward a Software-Based Network: Integrating Software Defined Networking and Network Function Virtualization", *IEEE Network*, vol. 29, no. 3, pp. 36-41, 2015.
- [10] R. Kumar, N. Gupta, S. Charu, K. Jain, S. K. Jangir, "Open Source Solution for Cloud Computing Platform Using OpenStack", *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 89 – 98, 2014.
- [11] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, "Network Function Virtualization: Challenges and Opportunities for Innovations", *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90-97, 2015.
- [12] R. Yu, G. Xue, V. T. Kilari, X. Zhang, "Network Function Virtualization in the Multi-Tenant Cloud", *IEEE Network*, vol. 29, no. 3, pp. 42-47, 2015.
- [13] R. Jain, S. Paul, "Network Virtualization and Software Defined Networking for Cloud Computing: A Survey", *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24-31, 2013.
- [14] Ceph, <http://docs.ceph.com/docs/master>, Accessed on March 2019.
- [15] E. Tasoulas, E. G. Gran, B. D. Johnsen, K. Begnum, T. Skeie, "Towards the InfiniBand SR-IOV vSwitch Architecture", *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM2015)*, Ottawa, Canada, pp. 10, May 2015.
- [16] P. Lothberg, "Terstream presentation", *RIPE 67 Meeting*, Greece, Athens, October 2013.