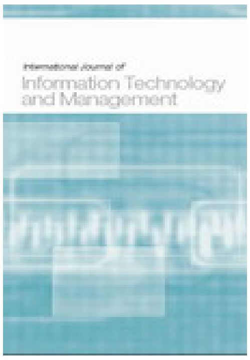


International Journal of Information Technology and Management

ISSN online: 1741-5179 - ISSN print: 1461-4111
https://www.inderscience.com/ijitm

OpenStack: a virtualisation overview

Faouzi Mechraoui, Pedro Martins, Filipe Caldeira

DOI: 10.1504/IJITM.2024.10061679

Article History:

Received:	01 May 2019
Accepted:	20 April 2020
Published online:	22 January 2024

OpenStack: a virtualisation overview

Faouzi Mechraoui

UCLL,
University of Leuven Limburg,
Leuven, Belgium
Email: faouzimechraoui@gmail.com

Pedro Martins* and Filipe Caldeira

CISeD – Research Centre in Digital Services,
Polytechnic of Viseu, Portugal
Email: pedromom@estgv.ipv.pt
Email: caldeira@estgv.ipv.pt
*Corresponding author

Abstract: The major cloud computing software companies offer a new concept, on which resources are virtualised to provide these as a service on the internet. Currently, there are multiple service providers, and additional options to virtualise services on-premises. OpenStack is an open-source alternative to create virtual local or cloud setups, which supports petabytes of data, unlimited scale, and configurable networking. These features make this tool suitable for large scale virtualisation, reducing maintenance costs and optimising hardware resource utilisation (e.g., schools, government). This paper presents an overview of the study of the OpenStack software, oriented to build a scalable hosting architecture suitable for an educational setup. Functional and architectural details are discussed to implement unique cloud computing to fit virtualisation purposes. An experimental virtualisation setup is described in the scope of an educational scenario. Finally, a guideline to configure OpenStack is given.

Keywords: OpenStack; infrastructure as a service; IaaS; virtualisation; cloud computing; open source.

Reference to this paper should be made as follows: Mechraoui, F., Martins, P. and Caldeira, F. (2024) 'OpenStack: a virtualisation overview', *Int. J. Information Technology and Management*, Vol. 23, No. 1, pp.1–12.

Biographical notes: Faouzi Mechraoui is a student and researcher at the University UC Leuven-Limburg, Leuven, Flanders, Belgium, he also learned the importance of informatics techniques to modern systems from experts. He has an extended knowledge, for his age, in the field, and travelled across Europe, looking for abroad experience. Recently he finished his Bachelor's degree, and he is progressing in his Master's degree in the topic of Systems Virtualisation. He hopes to intern in a cooperative department soon.

Pedro Martins is a Computer Science and Informatics specialist with a decade of successful experience in lecturing and researching. He specialises in databases, parallel and distributed computation, bigdata challenges, data visualisation, among other topics. He regularly attends several conferences in various fields and trends, such as distributed computation, databases, big data, security, image processing, networks, and so on. A strong believer in the power of positive thinking in the workplace, Pedro regularly develops software(s) to assist his work with effective performance, security and usability techniques.

Filipe Caldeira is a Professor at the Polytechnic Institute of Viseu, Portugal. He is a researcher at the Research Centre in Digital Services (CISeD) of the Polytechnic Institute of Viseu and at the Centre for Informatics and Systems of the University of Coimbra. His main research interests include ICT security, namely, trust and reputation systems, smart cities and critical infrastructure protection. His research papers were published in various international conferences, journals and book chapters. He has been recently involved in some international and national research projects.

1 Introduction

Technologies like web services, virtualisation, service oriented architecture (SOA), grid computing, and others, are brought together by the relatively new concept of cloud computing. This paradigm is also changing the way business models deliver I.T. as a service request, being scalable and supporting elasticity (Spanaki et al., 2018; Buyya et al., 2018). In general cloud computing is associated with outsourcing of resources, each one with a service level agreement (SLA) and mechanisms that allow on-demand pay-per-use billing (Schubert et al., 2010). Major cloud service providers include Amazon EC2, Microsoft Azure, Google Apps, and IBM cloud (Yang et al., 2017). OpenStack, VMWare ESXi, Proxmox, Ovirt, XenServer, among others, are IaaS virtualisation packages/operating systems that allow anyone to virtualise services and explore the hardware capabilities of each machine (memory, disk, CPU, network). Currently, OpenStack is the most adopted open-source virtualisation platform. Mostly because it is simple to implement as infrastructure (IaaS), and it is massively scalable (Couto et al., 2018). Many research studies focus on detailed comparisons of different virtualisation services (del Castillo et al., 2013; Bist et al., 2013). OpenStack also demonstrated success on education (del Castillo et al., 2013) and industry (Bonner et al., 2013; Campos et al., 2013). This paper makes an overview of an OpenStack architecture-oriented aspecific educational setup, showing the essential services that are required to be installed, sized, and configured (e.g., users, groups, limits, projects, templates, virtual networks).

The paper is organised as follows. First, Section 2 resumes the related work in this field. Then, the OpenStack architecture is presented in Section 3. After that, Section 4 describes and illustrates the experimental setup, in which OpenStack was implemented and tested. In Section 5 configuration steps with some examples are briefly described. Finally, Section 6 concludes the work and proposes possible future work.

2 Resumed related work

IaaS solutions concentrate on two aspects. First, the study of middleware platforms. Second, comparing studies of different approaches. Some examples of widely studied platforms are, Eucalyptus (Sotomayor et al., 2009; Nurmi et al., 2009), OpenNebula (Sotomayor et al., 2009; Milojičić et al., 2011), and Nimbus (Peng et al., 2009). In all previously mentioned works, several solutions and components are presented. Simultaneously there are several comparative studies of different solutions; some of them present an overview of solutions, others compare architectures and provide freedom of choice (Sempolinski and Thain, 2010; Peng et al., 2009). Regarding OpenStack, few works have addressed the solution itself (Mahjoub et al., 2011). However, this study is outdated, and OpenStack has evolved.

OpenStack is recent and under development with large potential and active support. The entire platform is licensed under Apache, and has these main characteristics:

- scalable: this solution is already deployed worldwide in companies whose data volumes are measured in petabytes of distributed architectures (Srinivasa and Santhosh, 2018)
- compatible and flexible: with support for most virtualisation solutions: ESX, Hyper-V, KVM, LXC, QEMU, UML, Xen and XenServer (Foresta et al., 2018; Endo et al., 2010)
- open: by using open-source technology, the entire code can be modified and adapted as needed.

Well-known business-oriented virtualisation packages such as VirtualBox, VMware Workstation, HyperV, Parallels, and others, are well fitted to virtualise operating systems. However, they are dedicated to small-scale scenarios. Additionally, there is VMware ESXi, OVirt, Proxmox (the last two, are open-source projects) which support a broader scale of virtualisation, with easy installation and configuration.

This paper focuses on the OpenStack architecture, application and configuration, oriented to a real (medium scale) virtualisation setup.

3 OpenStack architecture

The OpenStack architecture is built on top of three main components: compute, image, and object. Additionally, those components can be divided into six groups: computing, imaging, networking, storing, shared, and supporting services, which despite not being the core of OpenStack are essential for its operation (Mwaura, 2018). Figure 1 shows the architecture of OpenStack and how they interconnect.

3.1 Computing

The compute component known as Nova, Figure 1, consists of a management console that controls the infrastructure that controls the IaaS cloud. It can be compared to Amazon EC2 and RackSpace Cloud Servers. The Nova component allows managing large, redundant and scalable networks of virtual machines. It presents an interface and an API to administrate and orchestrate the cloud. Simultaneously it includes management instances for servers, networks, and access controls. The compute module has no hardware pre-requirements and is completely independent of the hypervisor.

Figure 1 OpenStack architecture components (see online version for colours)

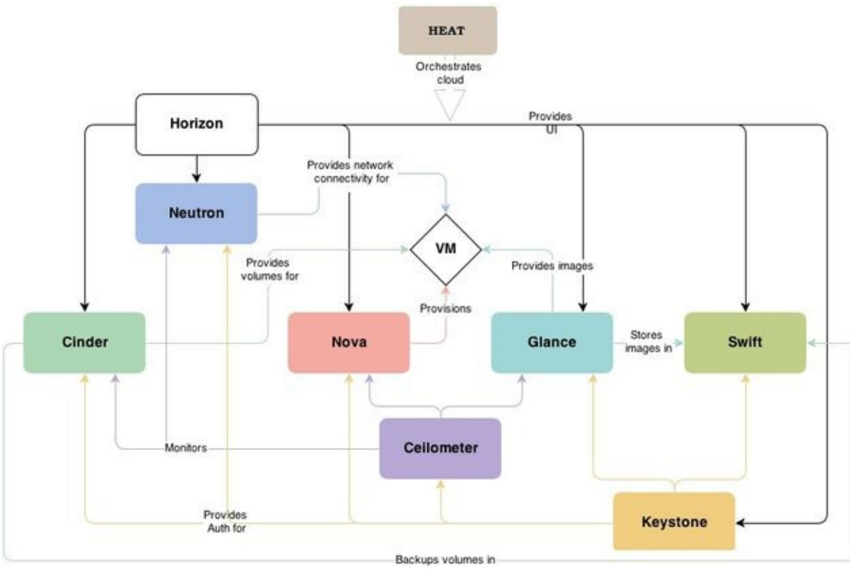


Figure 2 Nova architecture (see online version for colours)

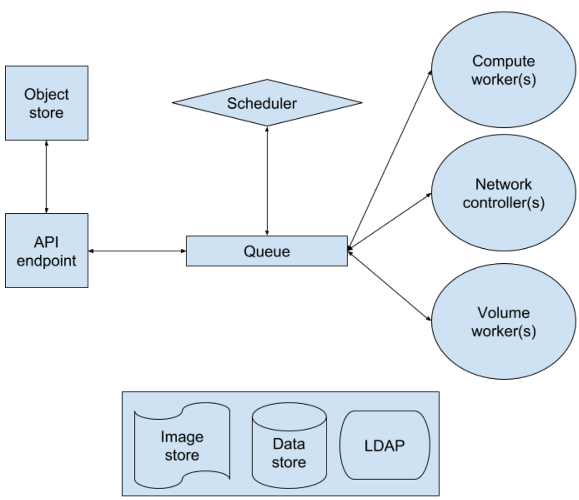


Figure 2 shows *Nova* compute seven internal architecture. This component provides virtual servers on demand. It interacts through the API module with the hypervisors while supporting several types such as KVM, Xen, VMware or Hyper-V. Simultaneously it is also capable of orchestrating running instances, networks, and access control.

First, the API server, which is the main component of Nova, works as a front-end service control for the hypervisor. The message queue acts as a message dispatcher for the exchanged instructions facilitating communication. The compute controller manages the life cycle of the created instances and is responsible for creating and manipulating the virtual servers. The object store offers storage services. The volume controller module handles the assignment and manipulation of volumes. The network controller is responsible for creating the bridges/VLAN/DHCP/DNS and firewall rules. At last, the scheduler is responsible for distributing tasks and deciding where they are executed.

3.2 Imaging service

Glance is the imaging service that allows lookup and retrieval of virtual machine images (storage, recording, distribution). Glance is used for discovering, registering and retrieving virtual machines through an API that allows querying virtual machine image meta-data, cataloguing and managing image service libraries. This service is an essential service for basic cloud architecture implementation.

3.3 Networking

Neutron is the OpenStack networking service that provides network connectivity as a service between the interface devices managed by other OpenStack services (e.g., *Nova*). Dynamic host configuration protocol (DHCP), static internal protocols (I.P.) and virtual area network (VPN), along with other typologies, are all the responsibility of the Neutron service.

This service allows users to take advantage of other frameworks from supported vendors (e.g., intrusion detection, load balancing, VPN).

3.4 Storing

OpenStack *Swift*, also known as object storage, is used to create storage space that is redundant and scalable, supporting multiple petabytes of data. Note that it is not a file-system, Swift is designed for long-term storage of large data volumes. The distributed architecture allows multiple access points to avoid a single point of failure (SPOF). Swift can be used by the Cinder component to back up virtual machines volumes.

OpenStack *Cinder*, also know as block storage or Cinder volumes, provides persistent block/volume storage to virtual machines. Together with Swift, Cinder can be used to create virtual machine backups and snapshots.

3.5 Shared services

The *Horizon* service is the OpenStack dashboard; a modular web application providing a user interface for cloud infrastructure management. This service interacts with all other services' public APIs, representing an essential service for basic cloud architecture implementation.

OpenStack *Keystone*, is an identity service for authentication and authorisation. Without Keystone, there is no compliance between services.

The *Ceilometer* service provides configurable parameters regarding CPU, memory, and network for all services involved in the platform. This service delivers a single point of contact for billing services.

3.6 Supporting services

The *database*, by default MySQL, is a relational database management system to store configurations. Typically it is present at the controller node.

The *advanced message queue protocol* (AMQP), is a messaging system used for inter-process communication to send and receive messages between processes, by default it uses RabbitMQ but others such as Qpid or ZeroMQ are supported as well. This queue also makes it possible to hide other components' whereabouts and definitions.

4 Experimental setup

This section describes a case in which an experimental setup is used to test the OpenStack virtualisation platform, oriented to an educational environment.

In an educational setting, two scenarios exist in which a *infrastructure as a service* (IaaS) solution may be used. First, students use virtual machines installed on physical machines belonging to the institution or their personal machines. Installing and configuring these may be cumbersome and time-consuming. Moving these activities to an IaaS system provided by the institution will ease deployment, usage and management of the virtual machines. The goal of this setup is building a proof-of-concept that would satisfy the requirements of both scenarios.

4.1 Scenario 1

In the first scenario, the requirements consist of giving the students a platform to complete their assignments on. Currently, students that need to complete tasks or projects generally use their equipment to set up and use virtual machines. However, the process of installing virtual machines can take quite some time, and some compatibility problems might surface during the virtual machine's life. These problems may be avoided by transitioning to a virtualisation platform based on campus. This platform needs to provide users with virtual machines without waiting for an installation to finish, and the virtual machines O.S. images should be screened beforehand by the administrators to ensure no compatibility problems might arise during usage.

Additionally, this gives the users to the possibility to complete their work from other devices if they so desire, the user interface of OpenStack is provided by the Horizon component which presents a web interface to manage the virtual machines and networks efficiently.

4.2 Scenario 2

For the second scenario, an environment is asked in which the virtualisation platform will replace the physical machines used in the classrooms on campus. The most important goals are making it easy and straightforward to manage the (virtual) tools and efficient distribution of system resources. Consequentially the physical machines could be replaced or repurposed as thin-clients which will connect to the OpenStack cluster and present one or more virtual machines through a remote desktop protocol such as VNC or SPICE.

4.3 Setup

For a proof of concept a simple OpenStack installation will be deployed to a single machine with following specifications:

- 64 GB memory
- 3.8 TB storage
- CPU: Intel Xeon E3-1225.

Each student will be associated to an account managed by *Keystone* and will have access to a *project*. Such a project serves as a room for the student to deploy his virtual machines(s) in isolation of the outside world and the other projects living in that cloud. A project can also be assigned to multiple students. This is useful for assignments where more students need to collaborate on the same setup.

Connectivity can be achieved within a project using a private network using private range of addresses. This network can be connected to another network using a virtual router provided by the OpenStack networking service *Neutron*. An admin network will be bridged to the physical (*provider*) network. This admin network will be made available to all projects. NAT will be applied using *floating* addresses as the public range. This range is the range used on the provider network.

The administrators will provide system images through the OpenStack image service *Glance*, and a working virtual machine can be spun up in under a minute time using this approach. If wanted, the disk of a virtual machine *instance* can be kept to be reused in another instance. This service will be provided by *Cinder*.

Furthermore, each department can be separated through *domains*. Supported by *Keystone*, a domain isolates users, project, administrators and networks. This is useful to separate different departments and use cases.

To ensure good performance for all users limits will be applied to the users of the OpenStack cloud. These limits can be defined on a per-project basis and can include several available instances as well as limits to the available resources and floating I.P. addresses. For this experiment, the limits will be set as such for each student project, assuming each student will generally need about five instances about any given time. Limits can be modified for each project individually if necessary.

- CPU: 2 vCores
- memory: 3 GB
- storage: 90 GB

5 Setup configurations

For this PoC the most common OpenStack services are being installed by PackStack, a package which uses Puppet modules.

Once the installation is complete, new domains are added from the CLI of the server. A file will be added to request a token from *Keystone* to be able to execute OpenStack admin operations. Admin users need to be added to manage each domain. For this *roles* are used. The default rules are *admin* and *member*. As roles are a relation between a *user* and a *project* an admin project needs to be added, and the 'admin' role needs to be assigned for the admin user of that domain to the project for the admin user to manage the domain from the web-interface.

For security reasons, multi-domain login is disabled by default for the web interface, and this can easily be enabled by editing the file */etc/OpenStack-dashboard/local_settings*. From then on the web-interface can be used to log in to a domain and add the users and projects and assign roles. The limits for each project can be changed here too, to minimise work it would be preferable to set the default limits before doing the projects, this limits will be applied to all projects added afterwards.

In this setup, users should not be able to upload, delete or modify images. This is achieved by changing the file */etc/glance/policy.json*, requiring the admin role on the domain to execute the *add_image, delete_image* and *modify_image* actions.

5.1 Test 1

To verify how much users the experimental setup can support a simple observation of the memory usage during operation is made with various numbers of instances. These instances are originating from the same Debian 8.5.0 installation which also has the XFCE lightweight desktop environment running. The amount of memory used will be observed in two ways: firstly by using the program 'free' provided with CentOS which takes into account the total amount of memory used on that host, including by the operating system itself. The second source of data will be OpenStack itself, reporting on the memory consumption of the compute host, which in this case is the only host running the whole stack.

Table 1 Table displaying memory usage in megabyte for a number of instances running

Amount of instances	*Total memory usage*	*Usage as reported by OpenStack*
0	9,728	0
1	10,240	2,048
2	11,162	4,096
10	15,872	20,480
20	19,968	41,882
26	22,938	54,579
28	24,166	59,187
33	26,726	69,734
43	39,322	90,112

The results provided by the system and by OpenStack are wildly different (Table 1 and Figure 3). The OpenStack tools only report the amount of memory granted to the instances according to the 'flavours'; these flavours are profiles that describe the number of resources to be given to an instance. In this case, each instance is assigned 2 GB. The reported memory consumption is 2 GB times the number of instances. The real usage of memory, as reported by the operating system, is only the amount of memory needed by the instance at the time, memory is not being reserved preemptively. As the average Debian installation requires less than 2 GB, it is possible to assign more memory to the instances then there is system memory available. This technique is called 'memory over-commitment'. If the system memory is exhausted, it would be expected that the host system would start swapping and that the performance of memory operations would decline significantly.

Figure 3 Test 1, a graphic representation of the memory usage (see online version for colours)

An additional test could be performed to compare performance in a setup where each instance is configured with a swap partition or file and a setup two where no swap mechanism would be enabled on the instances and where they would rely on the swapping capabilities of the host.

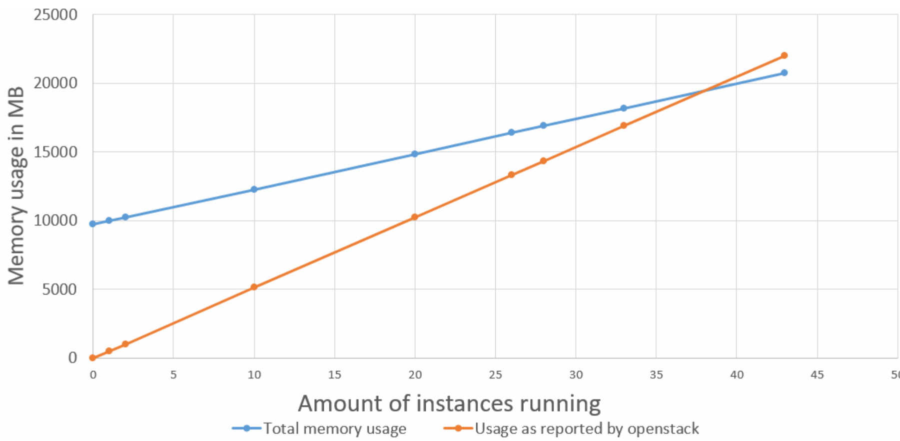
Table 2 Table consisting of memory usage in megabyte for a number of instances running

Amount of instances	*Total memory usage*	*Usage as reported by OpenStack*
0	9,728	0
1	9,984	512
2	10,240	1,024
10	12,288	5,120
20	14,848	10,240
26	16,384	13,312
28	16,896	14,336
33	18,176	16,896
43	20,736	22,016

5.2 Test 2

A second test is similar but would more closely simulate a real use case. Here the same test is performed but with Debian machines that do not come with a desktop environment, the assigned memory would be 512MB for each instance. This should be enough for typical use with only the command-line interface.

Figure 4 Test 2, a graphic representation of the memory usage (see online version for colours)



As expected (Table 2 and Figure 4), a better-optimised image results in a smaller memory footprint. The premise of this experiment is the same as the first one. Again OpenStack reports the memory that is assigned to the instances while that memory is not immediately dedicated, which makes over-committing memory feasible. The logical conclusion is that the better an instance image is optimised, the more users can be served on a given set of resources.

6 Conclusions and future work

The deployment of IaaS with OpenStack provides tools for creating and managing virtual machines on top of existing resources. The overview of OpenStack performed in this study, alongside with the experimental setup, and finally, the configuration setup, show that OpenStack follows the trends dictated either by the needs of users or by compliance with the new open standards.

In this paper, beside doing an overview of OpenStack virtualisation software, a useful test case applied to educational resource virtualisation is shown in its most relevant configurations.

Future work involving resource virtualisation point to benchmark tests to stress OpenStack, and to study how it manages load, available resources and scaling. Furthermore, the final goal would be an implementation of OpenStack to virtualise the entire on-premise educational system (Viseu Polytechnic Institute), allowing students to manage their limited number of instances.

Acknowledgements

This work is funded by National Funds through the FCT – Foundation for Science and Technology, I.P., within the scope of the project Ref UIDB/05583/2020. Furthermore, we would like to thank the Research Centre in Digital Services (CISeD), the Polytechnic of Viseu for their support.

References

Bist, M., Wariya, M. and Agarwal, A. (2013) 'Comparing Delta, Open Stack and Xen cloud platforms: a survey on open source IaaS', *2013 IEEE 3rd International Advance Computing Conference (IACC)*, pp.96–100, IEEE.

Bonner, S., Pulley, C., Kureshi, I., Holmes, V., Brennan, J. and James, Y. (2013) 'Using OpenStack to improve student experience in an HE environment', *Science and Information Conference (SAI) 2013*, pp.888–893, IEEE.

Buyya, R., Srirama, S.N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., Gelenbe, E., Javadi, B., Vaquero, L.M., Netto, M.A. et al. (2018) 'A manifesto for future generation cloud computing: research directions for the next decade', *ACM Computing Surveys (CSUR)*, Vol. 51, No. 5, p.105.

Campos, I., Fernández-del Castillo, E., Heinemeyer, S., Lopez-Garcia, A., Pahlen, F. and Borges, G. (2013) 'Phenomenology tools on cloud infrastructures using openstack', *The European Physical Journal C*, Vol. 73, No. 4, p.2375.

Couto, R.S., Sadok, H., Cruz, P., da Silva, F.F., Sciammarella, T., Campista, M.E.M., Costa, L.H.M., Velloso, P.B. and Rubinstein, M.G. (2018) 'Building an IaaS cloud with droplets: a collaborative experience with OpenStack', *Journal of Network and Computer Applications*, Vol. 117, pp.59–71.

del Castillo, J.A.L., Mallichan, K. and Al-Hazmi, Y. (2013) 'OpenStack federation in experimentation multi-cloud testbeds', *2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, Vol. 2, pp.51–56, IEEE.

Endo, P.T., Gonçalves, G.E., Kelner, J. and Sadok, D. (2010) 'A survey on open-source cloud computing solutions', *Brazilian Symposium on Computer Networks and Distributed Systems*, Vol. 71.

Foresta, F., Cerroni, W., Foschini, L., Davoli, G., Contoli, C., Corradi, A. and Callegati, F. (2018) 'Improving OpenStack networking: advantages and performance of native sdn integration', *2018 IEEE International Conference on Communications (ICC)*, pp.1–6, IEEE.

Mahjoub, M., Mdhaffar, A., Halima, R.B. and Jmaiel, M. (2011) 'A comparative study of the current cloud computing technologies and offers', *2011 First International Symposium on Network Cloud Computing and Applications (NCCA)*, pp.131–134, IEEE.

Milojičić, D., Llorente, I.M. and Montero, R.S. (2011) 'OpenNebula: a cloud management tool', *IEEE Internet Computing*, Vol. 15, No. 2, pp.11–14.

Mwaura, G. (2018) *Design and Implementation of OpenStack Hack-Lab*, Metropolia Ammattikorkeakoulu.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L. and Zagorodnov, D. (2009) 'The eucalyptus open-source cloud-computing system', *9th IEEE/ACM International Symposium on Cluster Computing and the Grid 2009, CCGRID'09*, pp.124–131, IEEE.

Peng, J., Zhang, X., Lei, Z., Zhang, B., Zhang, W. and Li, Q. (2009) 'Comparison of several cloud computing platforms', *Second International Symposium on Information Science and Engineering (ISISE 2009)*, pp.23–27, IEEE.

Schubert, L., Jeffery, K. and Neidecker-Lutz, B. (2010) *The Future of Cloud Computing: Opportunities for European Cloud Computing beyond 2010*, Expert Group report, Public Version 1.

Sempolinski, P. and Thain, D. (2010) 'A comparison and critique of Eucalyptus, OpenNebula and Nimbus', *2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, pp.417–426, IEEE.

Sotomayor, B., Montero, R.S., Llorente, I.M. and Foster, I. (2009) 'Virtual infrastructure management in private and hybrid clouds', *IEEE Internet Computing*, Vol. 13, No. 5, pp.14–22.

Spanaki, K., Gürgüç, Z., Mulligan, C. and Lupu, E. (2018) 'Organizational cloud security and control: a proactive approach', *Information Technology & People*.

Srinivasa, K. and Santhosh, V. (2018) 'An insight into OpenStack', *Design and Use of Virtualization Technology in Cloud Computing*, pp.243–259, IGI Global.

Yang, C., Huang, Q., Li, Z., Liu, K. and Hu, F. (2017) 'Big data and cloud computing: innovation opportunities and challenges', *International Journal of Digital Earth*, Vol. 10, No. 1, pp.13–53.