



INTERDEPARTMENTAL PROGRAMME OF POSTGRADUATE STUDIES

IN

INFORMATION SYSTEMS

MSc Dissertation

**INFRASTRUCTURE AS A SERVICE (IAAS) IN CENTRE FOR RESEARCH
AND TECHNOLOGY HELLAS (CERTH)**

By

IOANNIS KOUIMTZIS

**Submitted as a prerequisite in fulfillment of the requirements for the acquisition of
the postgraduate degree in Information Systems**

10/2018

Thessaloniki, Greece

Acknowledgements

This thesis is my master thesis for the Interdepartmental Programme of Postgraduate Studies in Information Systems of the University of Macedonia. In the context of its completion, I would like to thank my professors, Manos Roumeliotis and Kostas E. Psannis for the inspiration, help and support they have provided to me throughout its duration. Of course I also want to thank my parents, Andreas and Polyxeni, my sister Kyriaki and my aunt Stavroula for their patience, the support and impetus they have given to me throughout the whole postgraduate course.

Ioannis Kouimtzis

Thessaloniki, October 2018

Abstract

In this paper we explore whether and in what way Cloud Computing and particularly Infrastructure as a Service (I.a.a.S) can improve the services that Information Technology can provide in a research center and specifically in Center for Research and Technology Hellas (CERTH).

The characteristics that are being explored are the continuous availability of the services and the resources, the elasticity, which is the variation of them, the reduction of the costs and the savings of the energy that is consumed, the security and the probable locking in a vendor's services.

Next we present 3 implementations of consumer IaaS clouds, the Amazon's Elastic Compute Cloud (EC2), the Google's Compute Engine and the Microsoft Azure, as well the OpenStack, the most widespread open source (private or not) cloud development platform.

Key words: Cloud computing, Infrastructure as a Service, IaaS, research center, CERTH, public cloud, private cloud, community cloud, hybrid cloud, elasticity, availability, security, privacy, cost saving, energy saving, vendor lock-in.

CONTENTS

1	Introduction	1
2	The Cloud Computing in general	6
3	Criteria – Characteristics	10
3.1	Elasticity	10
3.2	Security – Privacy	12
3.3	Energy savings	16
3.4	Availability	18
3.5	Vendor lock in	21
4	Investigated IaaS platforms.....	23
4.1	Amazon EC2.....	23
4.2	Google Compute Engine.....	30
4.3	Microsoft Azure	36
4.4	OpenStack	43
5	OpenStack presentation.....	50
5.1	Security groups.....	56
5.1.1	New Security Group for SSH and ICMP (ping) in Linux VMs	58
5.1.2	New Security Group for RDP and ICMP (ping) in MS Windows VMs	62
5.2	Key Pairs	67
5.2.1	Creation:.....	67
5.3	Images	68
5.4	User management.....	76
5.4.1	User creation	77
5.5	Project management.....	79
5.5.1	Project creation	80
5.6	Client1 project.....	83
5.6.1	The creation of a network for client1	85
5.6.2	The launch of an instance	94
5.6.3	A floating IP allocation	109
6	Conclusions	114
7	References.....	118
8	Appentices.....	129

Images List

Image 1: Service model of Cloud: (a) Software as a service (SaaS), (b) Platform as a Service (PaaS), και (c) Infrastructure as a service (IaaS) (Deepak Puthal et al. (2015))	7
Image 2: Cloud service delivery models (Mohamed Al Morsy, John Grundy and Ingo Müller (2016))	8
Image 3: Cloud solutions based on the system’s deployment and service model (Deepak Puthal et al. (2015))	8
Image 4: Multi-tenancy approaches (Mohamed Al Morsy, John Grundy and Ingo Müller (2016))	12
Image 5: Server Virtualization in Cloud Computing (Amani S. Ibrahim, James Hamlyn-Harris, John Grundy (2016))	14
Image 6: The relationship between different types of storage	24
Image 7: Availability zones and regions	26
Image 8: The installation of DevStack is complete	51
Image 9: Horizon log in screen	53
Image 10: Sign in as admin	53
Image 11: The initial projects installed by DevStack	54
Image 12: The initial network topology created during installation for the project “demo”	55
Image 13: The Security Groups with the preinstalled default one	56
Image 14: The rules of the “default” security group	57
Image 15: The creation of a new security group for the Linux VMs. It allows SSH connections and ICMP (ping) communication	58
Image 16: The initial rules of the new security group	59
Image 17: Addition of the rule for SSH connections	59
Image 18: Addition of the rule for ICMP inwards	60

Image 19: Addition of the rule for ICMP outwards	61
Image 20: The final rules of the security group for the Linux VMs	62
Image 21: The creation of a new security group for the MS Windows VMs. It allows Remote Desktop connections and ICMP communication	62
Image 22: The 3 security groups	63
Image 23: Addition of the rule for RDP connections	63
Image 24: Addition of the rule for ICMP inwards	64
Image 25: Addition of the rule for ICMP outwards	65
Image 26: The final rules of the security group for the MS Windows VMs	66
Image 27: The screen for key pair management with no key pairs created	67
Image 28: The creation of a key pair for SSH connections	67
Image 29: The first key pair shown in the key pair management screen	68
Image 30: The images management screen with the preinstalled cirros image...	68
Image 31: First screen of the upload of a new image	69
Image 32: Second screen of the upload of an image. It will be Ubuntu 16.04 server 64 bit LTS. We select QCOW2-QEMU emulation, AMD64 architecture and we define it as Public to be generally shared and with no protection	70
Image 33: First screen of the available metadata for the image. There are many options available	71
Image 34: Second screen of the available metadata for the image	72
Image 35: Third screen of the available metadata for the image	73
Image 36: Another upload of an image. It will be MS Windows Server 2012 R2 Standard evaluation. We select QCOW2-QEMU emulation, x64 architecture and we define it as Public to be generally shared and with no protection	74
Image 37: We can again select the various metadata	75
Image 38: The user management screen with the predefined users	76

Image 39: We create a new user named cl1_user for the client named client1. We define a password and can select the project the new user will be applied to	77
Image 40: We see the new user in the user management screen	78
Image 41: The project management screen with the predefined projects	79
Image 42: First screen of the creation of the new project named client1	80
Image 43: We select the users that will be members of the new project	81
Image 44: We select the groups of the users that will be members of the new project	82
Image 45: The project management screen with the new project shown	82
Image 46: We sign in as the new user cl1_user for the new project client1...	83
Image 47: The overview of the project client1	84
Image 48: The initial network topology of the new project client1. We see the same network named “public” that was created during installation	85
Image 49: The first screen of the creation of a new network named “private” for the client1 project	86
Image 50: The definition of the subnet of the private network	87
Image 51: The definition of the subnet details. We define the IP range, we enable DHCP for the range and the absence of a DNS server	88
Image 52: The new network with the new subnet for the client1 project is shown in the network topology	89
Image 53: First screen of the creation of a new router named router2 for client1 project. It will connect the new “private” network of the project with the “public” one of the infrastructure	90
Image 54: Overview of the new router	91
Image 55: We have not added any interfaces yet	91

Image 56: We add a new interface to the router router2. It will provide connection with the internal private network	92
Image 57: The new interface is shown	92
Image 58: The network topology with the 2 networks (public and private) connected via the router	93
Image 59: Overview of the client1 project	94
Image 60: The instances screen. We haven't created any yet	94
Image 61: First screen of the creation of a new instance	95
Image 62: Second screen of the new instance creation. We use the light cirros image. We want a new volume for disk space to be created that will be deleted if we delete the instance	96
Image 63: We select the m1.tiny flavor among the many available	97
Image 64: The only available private network is preselected	98
Image 65: We haven't defined any network ports.	98
Image 66: We select the security group for Linux VMs that we want	99
Image 67: We select a key pair to be used for SSH connections	100
Image 68: We have no special configuration script for the instance	101
Image 69: We haven't created any server groups to launch the instance in	102
Image 70: We could have defined scheduler hints for this instance if needed	103
Image 71: We can use many instance metadata for special configuration of the instance	104
Image 72: The new instance is being built	105
Image 73: The new instance is displayed at running state	105
Image 74: An overview of the instance with its private IP address displayed	106

Image 75: Overview of the instance continued	106
Image 76: The private IP address of the instance is displayed as up and active	107
Image 77: No problems recorded in the instance console log	107
Image 78: The console of the instance	108
Image 79: The action log of the instance	108
Image 80: We can associate a floating IP address to the instance from the instance overview screen	109
Image 81: We add a floating IP address to the instance we have created	109
Image 82: We select the public network to be the pool that will give the floating IP address and give a description to it	110
Image 83: Here we see the floating IP address and associate it to the instance	110
Image 84: We can see the floating IP address along with the private one in the overview of the instance	111
Image 85: The screen with the floating IPs of the project	112
Image 86: The network topology of the project with the instance shown	113

List of Tables

Table 1: Evaluation of the 4 IaaS solutions that are presented concerning the 5 characteristics/criteria that are explored	49
---	----

1 Introduction

The Centre for Research and Technology-Hellas (CERTH) is a research center in Greece and has its headquarters as well as the main body of its facilities in Thessaloniki, in the region of Macedonia. It is a legal entity governed by private law with non-profit status, supervised by the General Secretariat for Research and Technology (GSRT) of the Greek Ministry of Education, Research and Religious Affairs.

Today CERTH includes the following five institutes:

- **Chemical Process & Energy Resources Institute (CPERI)**
Sustainable & Clean Energy, Environmental Technologies, Chemical & Biochemical Processes, Advanced Functional Materials
- **Information Technologies Institute (ITI)**
Informatics, Telematics and Telecommunication Technologies
- **Hellenic Institute of Transport (HIT)**
Land, Sea and Air Transportation as well as Sustainable Mobility services
- **Institute of Applied Biosciences (INAB)**
Agri-biotechnology, Health Translational Research, Informatics for big bio-data
- **Bio-economy and Agro-technology Institute (IBO)**
Agrotechnology, Mechatronics, Biomedicine and Kinesiology

Today more than 700 people work at CERTH.

<https://www.certh.gr/5B4D1A98.en.aspx>

Every lab (consequently every institute) has its own financial resources, from research programs in which it is a partner, from contracts with private companies etc. This means that every lab occasionally can and/or is obliged to implement its own independent IT infrastructure according to its needs. At every occasion common infrastructure are the computer network, the connection with the Internet and the phones. Many labs use High Performance Computers and sometimes Storage Area Networks (SANs, storages). There isn't a central infrastructure capable to serve many labs, neither concerning processing power nor storage. The only step forward the last few years is that most servers are implemented as virtual machines, usually 4-5 per

physical server. Lastly some services of the Greek Research and Technology Network (GRNET) are being exploited, such as the high bandwidth interconnection between the universities and the research centers of Greece and the Internet, the provision of virtual machines and storage, safe videoconferences and High Performance Computing.

Scientists have the ability to connect to the internal network of CERTH using VPN connections and their personal or the lab computers from everywhere. There are also some www (World Wide Web) applications as MyCERTH that implements the bureaucratic procedures and collaborates with the ERP (Enterprise Resource Planning) program.

There are many papers about the adoption of Cloud Computing, but most of the times they are about companies and the cost reduction that they can have (Maricela-Georgiana Avram (Olaru) (2013)), (Tiago Oliveira, Manoj Thomas, Mariana Espadanal (2014)) or the public sector and the protection of sensitive personal data, as are the hospitals (Jiunn-Woei Lian, David C. Yen, Yen-Ting Wang (2013)).

Christos Stergiou, Kostas E. Psannis, Brij B. Gupta and Yutaka Ishibashi (2018) examine the security and privacy issues that arise when Cloud Computing integrates with Internet of Things as a base scenario for Big Data. An installation of a security “wall” between the Cloud Server and the Internet is proposed. Christos Stergiou, Kostas E. Psannis, Theofanis Xifilidis, Andreas P. Plageras and Brij B. Gupta (2018) survey how the combination of Social Networking and Big Data in a Cloud environment can improve security and privacy issues as excessive data and information exchange takes place. An efficient algorithm for advanced scalable Media-based Smart Big Data (3D, Ultra HD) on Intelligent Cloud Computing systems through wireless communications and networking is proposed by Kostas E. Psannis, Christos Stergiou and B. B. Gupta (2018). Improvement of Cloud Computing security with algorithms that can provide more privacy in the data related to Big Data technology is presented by Christos Stergiou and Kostas E. Psannis (2017). Ahmad M. Manasrah and Hanan Ba Ali (2018) propose an algorithm that offers cost reduction and load balancing of the tasks over the resources in cloud computing environments. José I. Benedetto, Guillermo Valenzuela, Pablo Sanabria, Andrés Neyem, Jaime Navón and Christian Poellabauer (2018) introduce a new code offloading framework that can extend the capabilities of mobile devices by migrating processor-intensive tasks to resource-rich substitutes. Xun Wu (2018) proposes a novel service selection and recommendation model to help users of mobile devices take advantage of mobile cloud computing and the services it offers.

Christos Stergiou, Kostas E. Psannis, Byung-Gyu Kimb and Brij Gupta (2018) examine the security issues and the benefits of the integration of Internet of Things and Cloud Computing. Christos Stergiou and Kostas E. Psannis (2016) survey if the benefits of Mobile Cloud Computing and Internet of Things can improve the use of the Big Data Applications. How Cloud Computing can help virus detection is shown by Vasileios A. Memos and Kostas E. Psannis (2014). K.E. Psannis*, S. Xinogalos and A. Sifaleras (2014) examine how the advantages of Mobile Cloud Computing can alleviate the limitations of Internet of Things.

Victor Chang, Yen-Hung Kuo and Muthu Ramachandran (2015) present a cloud computing adoption multilayered security for business clouds. It uses firewall, identity management and encryption for the files. Rostyslav Zabolotnyi, Philipp Leitner, Waldemar Hummer and Schahram Dustdar (2015) introduce JCloudScale, a Java-based middleware for building elastic applications in IaaS clouds, in order to avoid PaaS and the possible vendor lock-in, having full control of the used virtual machines. A technical debt-aware learning approach for autonomous elasticity management to address the gap between the ideal and actual resource provisioning is proposed by Carlos Mera-Gomez, Francisco Ramirez, Rami Bahsoon and Rajkumar Buyya (2017). A cloud-based elastic solution for VoIP providers implemented on OpenStack is presented by Victor Gonzalez Chamorro, Carlos Nunez Castillo and Fabio Lopez-Pires (2016). Kaveh Razavi Gerrit Van Der Kolk and Thilo Kielmann (2015) propose prebaked μ VMs, that is snapshots of minimal VMs that can boot in less than a second and can help the elasticity of application like Web servers.

Dan Gonzales, Jeremy M. Kaplan, Evan Saltzman, Zev Winkelman, and Dulani Woods (2015) present a cloud architecture security reference model and Cloud-Trust, a cloud security assessment model to assess a cloud computing system or a cloud service provider. Amir Teshome, Louis Rilling and Christine Morin (2018) describe the necessity of extending SLAs to include security monitoring terms and propose and apply a strategy on the case of network Intrusion Detection Systems. A secure cloud architecture with a hardware security module which isolates cloud user data from domains or cloud administrators is presented by Jinho Seol, Seongwook Jin, Daewoo Lee, Jaehyuk Huh, and Seungryoul Maeng (2016). Nidal Hassan Hussein and Ahmed Khalid (2016) survey cloud computing security challenges and propose a model for cloud security. Flora Amato, Francesco Moscato, Vincenzo Moscato and Francesco

Colace (2017) show how Model Driven Engineering Techniques can improve security analysis of cloud infrastructures.

Fei Teng, Lei Yu, Tianrui Li, Danting Deng and Frédéric Magoulès (2016) study batch-oriented consolidation and online placement for reserved virtual machines (VMs) and on-demand VMs on physical servers to save energy. Pliant system-based virtual machine scheduling approaches for reducing the energy consumption of cloud datacenters are proposed by A. Kertesz, J. D. Dombi and A. Benyi (2015). Syed Hamid Hussain Madni, Muhammad Shafie Abd Latiff, Yahaya Coulibaly, and Shafi'i Muhammad Abdulhamid (2016) study the recent issues of resource scheduling in IaaS cloud computing environment. Sangdo Lee Hyoungyill Park and Yongtae Shin (2012) survey multi-clouds which is “cloud-of-cloud” or rain computing, a cooperation of simple clouds that provides accessible resources to deal with the big data services. Panagiotis Kokkinos, Dimitris Kalogeras, Anna Levin and Emmanouel Varvarigos (2016) study the virtual machine live migration and disaster recovery considering long-distance networks between data centers.

L. Tomás, P. Kokkinos, V. Anagnostopoulos, O. Feder, D. Kyriazis, K. Meth, E. Varvarigos and T. Varvarigou (2017) present the Disaster Recovery Layer (DRL), a disaster recovery framework for OpenStack-managed datacenters. Security. Privacy and disaster recovery issues in the eHealth cloud domain are surveyed by Aqeel Sahi, David Lai and Yan Li (2016). Long Wang, Richard E Harper, Ruchi Mahindru, and Harigovind V Ramasamy (2016) present disaster protection and recovery for enterprise applications both at IaaS and PaaS level. Justice Opara-Martins, Reza Sahandi and Feng Tian (2016) analyze the vendor lock-in problem in the cloud environment, the necessity of the interoperability and portability of applications and of standardised formats and protocols. Eslam G. AbdAllah, Mohammad Zulkernine, Yuan Xiang Gu and Clifford Liem (2017) propose TRUST-CAP, a generic trust model for cloud-based applications with 4 components, integrity, access control, availability, and privacy to ensure security against common attacks.

In this paper we want to investigate how Cloud Computing can offer solutions for the –continuously changing- needs of a research center, possibly reducing the operating cost and giving the ability financial resources to be used elsewhere, besides the IT infrastructure.

Its objective is the form of the cloud that must be used.

But its objective also is the examination of certain commercial clouds and cloud development platforms, the analysis of the positive and the negative factors of every proposal and their parallel presentation so the Center can have the whole picture of the alternative solutions.

In the second chapter there is a brief presentation of the Cloud Computing. The services that Cloud Computing provides are mentioned, as well as the categories of clouds we see, depending on who provides them and how many distinct clouds we use at the same time. As can be seen from the analysis, the service that will be the subject of this study is Infrastructure as a Service (IaaS).

The third chapter presents a bibliographic presentation of the features that Infrastructure as a Service requires today. These features include elasticity, security, cost-efficiency and energy savings, continues availability and avoidance of a possible lock-in the provider due to politics or non-use of open technologies. These are the criteria by which the proposed clouds will be evaluated.

The fourth chapter presents and examines the proposed 4 most prevalent clouds and their corresponding service. These are 3 commercially available, Amazon's Elastic Compute Cloud (EC2), Google's Compute Engine and Microsoft Azure, and the most popular private / owned cloud development platform, OpenStack.

In the fifth chapter there is a short presentation of the management of the OpenStack platform using the DevStack simulation. Security management is introduced through the creation of firewall rules for remote management of Linux virtual machines through SSH and Windows virtual machines through Remote Desktop, image management for the creation of the virtual machines with corresponding operating systems, user / administrator creation, and the creation of a project, usually one per client. In the new project, we have the creation of a separate network, the installation of a virtual machine through an image and its interconnection with the client's network in the cloud.

2 The Cloud Computing in general

According to the US National Institute of Standards and Technology (NIST), cloud computing provides convenient, on-demand network access to computing resources (e.g., networks, servers, storage, applications and services) that are shared for these purposes and can be configured according to customer requirements while these resources are able to be provided and removed with minimal involvement of administrators or the provider (Deepak Puthal et al. (2015); (Mohamed Al Morsy, John Grundy and Ingo Müller (2016))).

Three types of cloud computing services can be distinguished: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

In Software as a Service (SaaS), the client has the ability to use some applications over the World Wide Web without installing anything on its computer beyond a simple web browser. He can't and doesn't need to know the infrastructure behind the application (servers, networks, operating systems, programming languages, etc.).

In Platform as a Service (PaaS), the client has the ability to develop its own applications in the cloud using the specific tools and infrastructures provided by it. This is the platform, it involves strictly specific tools. Customers can't use others. Again, the customer is not related to the infrastructure hidden behind, the only thing he controls is his own application.

Finally, Infrastructure as a Service (IaaS) provides computing resources to the client such as processing power, storage media, network and other lower level resources comparatively to PaaS to create his own virtual machines (VMs) such as servers, PC clients etc.; to install any operating system he wants in them and any programming tools and applications he wants and needs. In all of these he has complete control. He does not have control to everything hidden underneath (physical servers, physical networks, etc.) (Deepak Puthal et al. (2015)).

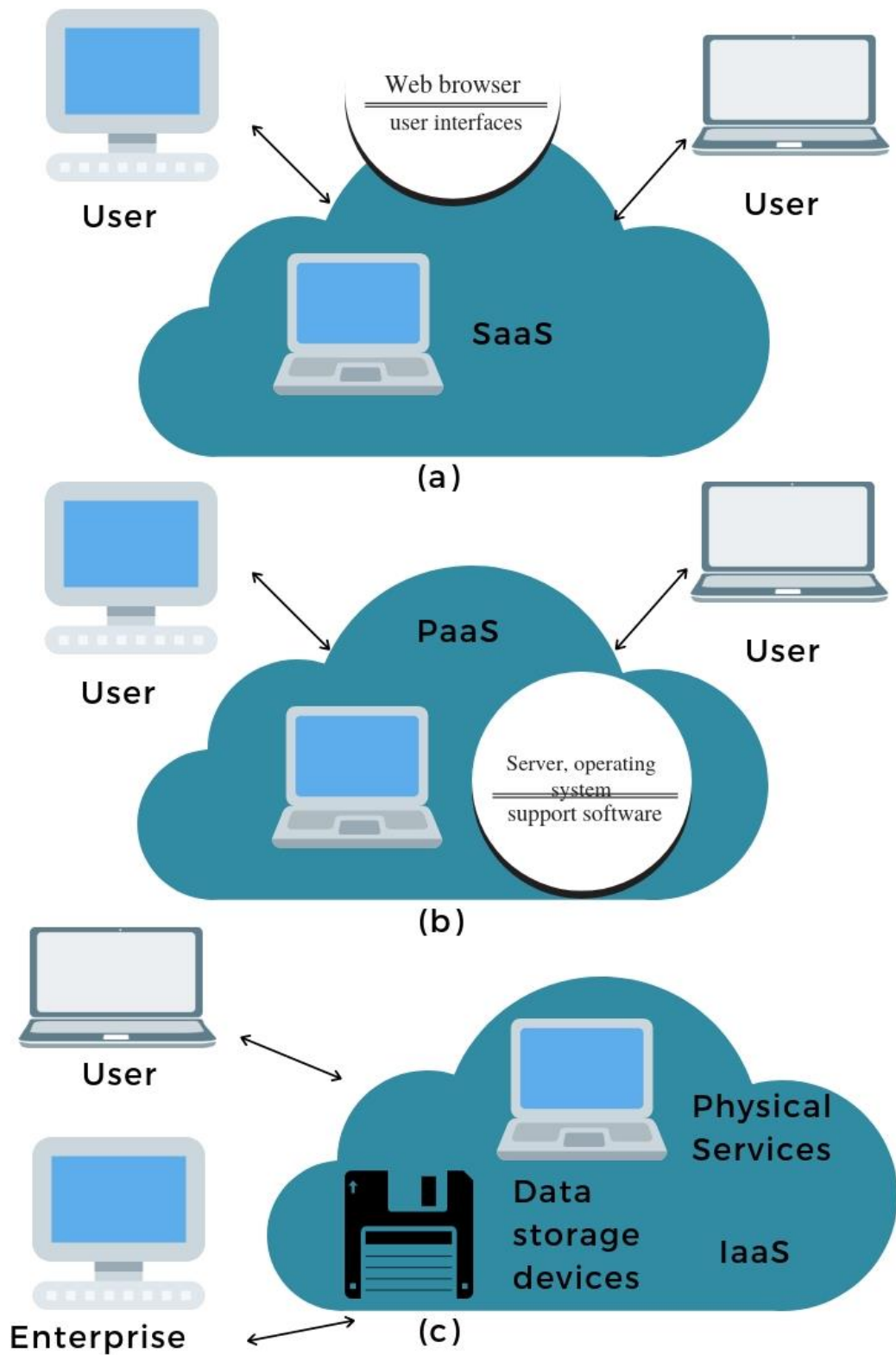


Image 1: Service model of Cloud: (a) Software as a service (SaaS), (b) Platform as a Service (PaaS), και (c) Infrastructure as a service (IaaS) (Deepak Puthal et al. (2015))

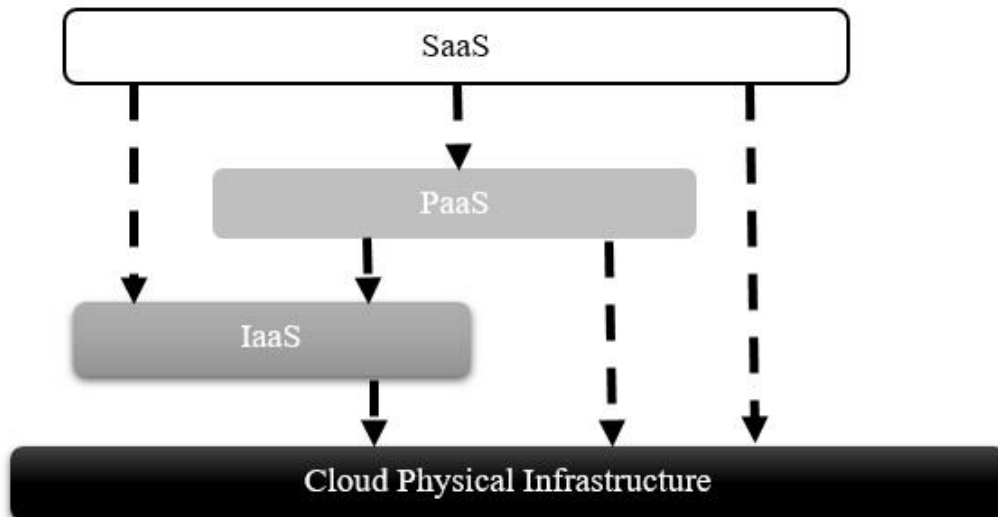


Image 2: Cloud service delivery models (Mohamed Al Morsy, John Grundy and Ingo Müller (2016))

There are four cloud computing categories, the private cloud, the public cloud, the community cloud, and the hybrid cloud.

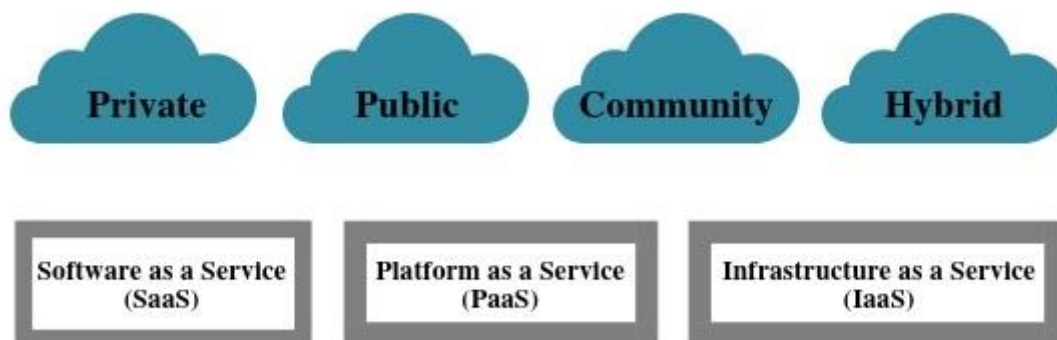


Image 3: Cloud solutions based on the system's deployment and service model (Deepak Puthal et al. (2015))

In the private cloud the client implements (with own resources or with the help of a third company) his own cloud, which belongs exclusively to him and only he has access to it. He buys the infrastructure that he installs at his premises and implements the cloud with his own resources. This means increased implementation and maintenance costs but also increased security.

Public cloud is usually implemented by very large companies that implement the necessary infrastructure to provide services to anyone. Anyone can rent services without worrying about what's behind them. This form of cloud is the cheapest for the customer but many security issues arise because of access to the infrastructure by many factors (companies, administrators, etc.) and easy network access.

Community cloud is a form of private cloud that implement and share similar organizations with a similar objective and needs that lead them to similar cloud characteristics. This reduces development and maintenance costs.

Some or all of the above forms coexist in the hybrid cloud. The client uses whatever formats are convenient, by mixing any of them he particularly wishes. In this way, he has the ability to transfer his applications to whatever cloud he wants (from any of them he has access to) dynamically, depending on the power, security and privacy that he needs (Deepak Puthal et al. (2015).

3 Criteria – Characteristics

3.1 Elasticity

A key characteristic of the clouds that both customers and providers are looking for is elasticity. Elasticity enables the system to add or remove computing resources automatically according to the load so that resources can respond to demand as quickly as possible (Nikolas Roman Herbst et al. (2015)).

Providers want elasticity so as to take advantage of their infrastructure at any time to the fullest extent by reducing operating costs while customers want elasticity in order not to be charged unnecessary resources according to the load of their applications. In this way customers have the impression that the resources of the cloud are inexhaustible.

The elasticity is distinguished to vertical and horizontal. In vertical elasticity, resources such as processing power, memory, and storage can be added or removed in real time to a particular virtual machine. In horizontal elasticity the number of virtual machines provided in an application is increasing or decreasing according to the requirements. These capabilities lead to increased performance and cost savings since the customer is not required to ask his provider for the maximum resources that he is likely to need and even from the beginning (such as in grid computing), but uses (and is charged) any of them he needs at any moment (Guilherme Galante et al. (2016)).

Methods to achieve elasticity are time scheduling, if we know when we will need more or fewer resources, and resource scaling based on thresholds set by the user. Thresholds can be percentages or specific amounts of processing power, memory, bandwidth, storage, etc. If a threshold is over or under, resources are added or removed (Yazhou Hu, Bo Deng, Yu Yang, Dongxia Wang (2016)).

This scaling method is auto-scaling. If e.g. 90% of the processing power is exceeded for more than one minute, the resources are increased. However, there is a risk of over-provisioning.

Another method is scaling-up and scaling-down. There are different types of virtual machines, with varying (staggered) power, memory, space, etc. Depending on

demand, a different type of VM is provided. The configuration and integration of different VMs leads to delays and consequently to over- (or under-) provisioning.

The most common method is scaling out (and in). There are same types of virtual machines that are very easy to provide. It leads to less over- (or under) provisioning and is what is preferred by most public clouds (Kai Hwang et al. (2015)).

The characteristics of the elasticity that are evaluated and characterize the various implementations of the clouds are the accuracy (or precision) and the time that the resources are adjusted. Accuracy is the amount of resources that becomes over- (or under) provisioned inevitably. Because no one customer wants to have under-provisioning, providers are looking to provide adequate resources so that only over-provisioning takes place. Their further concern is to make the least possible over-provisioning so that charges will be lower and attract customers. Time is both the net time it takes to provide the resources and the one that is misplaced in more or less resources when adjusting.

3.2 Security – Privacy

Cloud Computing is one of the most important developments in the field of Information Technology. Most people now recognize its strengths, but there is something that worries everyone and is the biggest inhibiting factor in its spread. Security.

The biggest change is the location of the data. Till now everyone is accustomed to having his data on his own computers and to be responsible for their own safety. Now, if people adopt the model of the public or hybrid cloud, they are called upon to trust their data in another organization-company. And lose control of the whole management.

At the same time, they must accept that their data and virtual machines will co-exist on the same infrastructure as others', and possibly their competitors'. This is the feature of multi-tenancy that hides many dangers. There are many implementations of multi-tenancy. The safest is when the client has his virtual machines tailored to its requirements and exclusively made for it, or again exclusively made but with configurations selected by a group already ready from the cloud. Less secure is when the same machine is shared by lease or depending on the customer's load requirements. In these implementations all the data co-exist in the same machine at the same time, sharing the same processors, the same memory, the same storage, etc. The risks of interception and leakage are enormous.

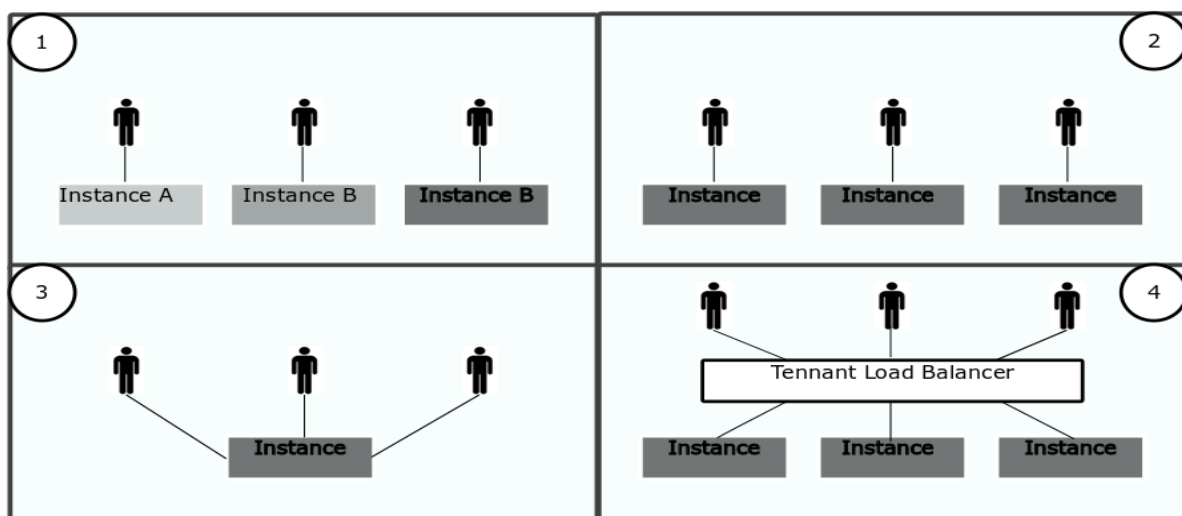


Image 4: Multi-tenancy approaches (Mohamed Al Morsy, John Grundy and Ingo Müller (2016))

To ensure secure multi-tenancy, there must be isolation of each client's data (either simply stored, processing or transporting) and location transparency. That is, it is not obvious where the data are located, not only at the server and storage level, but even at the country or continent level. IaaS isolation must also exist at the level of computing resources.

Like the multi-tenancy of own resources, elasticity, i.e. the commitment and release of additional resources according to the requirements of each client's applications, helps to manage resources more efficiently, limiting provider and customer costs. However, the provision of the same resources from the previous customer to the next (in particular memory and storage) hides the risk of information leakage. Managing resources also means a list of available resources at any time and of the customers they were given. This is another possibility of malicious attack.

In general, there is a lack of trust between providers and customers of clouds. Providers do not trust their customers because they do not know what applications they are running on the provided virtual machines, and customers do not trust providers because they do not know how infrastructure is structured and how effective are the protection methods from malicious attacks they have implemented. This is why Service Level Agreements (SLAs) must be implemented after negotiations that seek to safeguard the interests of both sides (Mohamed Al Morsy, John Grundy and Ingo Müller (2016)).

At resource level, clouds are based on virtualization technology. This technology enables many virtual machines to run different or many similar operating systems on the same physical server. This makes easier to manage server resources more efficiently. Till now we have many servers that use minimal resources. Now we can have one that is approaching 100% resource consumption, reducing costs far. Usually on a server hosting virtual machines we see Hypervisor, the basic software that manages machine resources, vSwitch that is a virtual network switch and through real network cards provides a network on the virtual machines and finally the virtual machines themselves. Each of these elements is a target of attacks. Virtual machines can be protected just like the real ones. The extra risk is due to the fact that they are vulnerable even when they do not work, as their image still exists in the cloud storage. The difficult part is to protect Hypervisor and vSwitch, as the eventual intrusion will affect all services that run or are going to run not only on the physical computer but throughout the cloud, creating a passage that provides access to the whole system.

Attackers install their own hypervisor or modify the existing ones. This way they control service provision and can create their own VM templates that will lead to the creation of VMs by customers controlled by them. In vSwitch the risks are the same as those of physical switches, perhaps smaller ones, since one can't connect a computer to a vSwitch like to a physical one and take control of the machine (Amani S. Ibrahim, James Hamlyn-Harris, John Grundy (2016)).

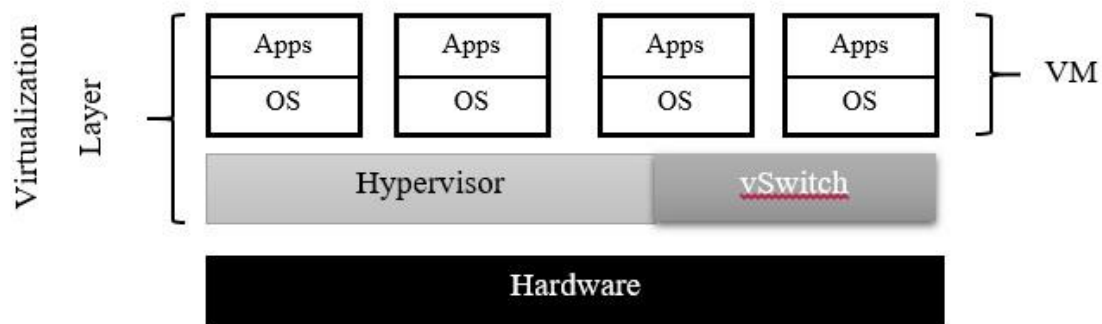


Image 5: Server Virtualization in Cloud Computing (Amani S. Ibrahim, James Hamlyn-Harris, John Grundy (2016))

Very important for the security of any implementation of a cloud is the security of the Cloud Management Layer (CML). This is the core that hosts and coordinates all services. It offers SLA management, service monitoring, IaaS, PaaS and SaaS, elasticity, charge and cloud management. Anyone who gains access at this level acquires administrator privileges and can do everything. Customers have access to CML through some Application Programming Interfaces (APIs) provided to them.

Access to the cloud is done through web browsers and web applications - HTTP / HTTPS protocols, web services and APIs - SOAP, REST and RPC protocols, or remote connections for the VMs and the IaaS storage - VPNs and FTP. In all cases authentication and authorization of users are required.

This leads to Identity and Access Management. Identity may involve users, services, machines, etc. It contains a set of information that characterizes a cloud component uniquely without, of course, important personal information. To achieve this, there are several standards such as SPML, SAML, OAuth and XACML.

Privacy is another big issue in the clouds. It is mainly achieved by encryption. But secure creation, storage, access, and exchange of keys is very difficult.

One last factor affecting security is the consumption of resources for security itself. Excessive effort for security leads to a great deal of resource consumption, and a compromise must also emerge here (Mohamed Al Morsy, John Grundy and Ingo Müller (2016)).

3.3 Energy savings

The components of a data center -that is, a place where the IT infrastructure (servers, switches, etc.) is hosted- that are energy-consuming are mainly Central Processing Units (CPUs), memory, storage, and network components. Of all these, clearly more energy is consumed by CPUs. It is very common in traditional environments to waste resources when a server with significant processing power offers a service of great importance but less demanding in power (e.g. domain controller, DNS server, etc.). CPU utilization is often seen less than 10%. The same can happen with memory, storage and network. A solution to this problem is given by the virtualization technology used in IaaS. The goal is, if possible, to approach 100% resource use. By migrating the VMs from one server to another server even when they are at running state, this becomes more possible (Soumya Ranjan Jena, V. Vijayaraja and Aditya Kumar Sahu, (2016)).

IaaS providers try to achieve a reduction in energy consumption through consolidation of services and hence the use of resources at the highest possible rate. Effective provision of services must be achieved using the minimum number of servers, which leads to a reduction in energy consumption.

With virtualization, multiple services and applications can be run simultaneously on the same server in isolation (theoretically). Through VMs, many different environments and operating systems can coexist on the same server. According to Popek and Goldberg, a virtual machine is an effective and isolated copy of a real computing engine. In order to be able to accommodate the many and sometimes different operating systems of the VMs on a server, we initially install a special software called hypervisor. The hypervisor communicates with the hardware and simulates virtual hardware that VMs communicate with. The simulation can be complete or exploit the potential for virtualization of new processors assisted by them and thus achieving better performance.

An alternative to virtual machines is containers. The containers are run in the kernel of some operating systems. This kernel can handle isolated virtual sub-machines. Because services running in containers communicate directly with the operating system instead of the hypervisor being between them and having double command movement, we have a much more efficient management of resources and especially memory. For

this reason, lately containers are becoming more preferable than the classic VMs. Of course, the isolation achieved is inferior to classic VMs, so it is likely to be easier to raise security issues on clouds based on them (Ismael Cuadrado-Cordero, Anne-Cecile Orgerie, Jean-Marc Menaud (2017)).

3.4 Availability

Availability is another feature that cloud providers always advertise. For their clients it is not just desirable but generally considered for granted. The usual hitherto practice is that applications run on a computer that regardless of power can't guarantee the continuous execution of applications as failures always occur. On the contrary, there is a point of view that in clouds, due to their abstraction in terms of infrastructure, applications will continue to run at all times since the cloud "always exists".

Data centers are designed to provide high availability. IaaS uses such centers and in this way it can provide computing resources promising high availability. In Service Level Agreements (SLAs) signed between the provider and the customer, the availability is reported as downtime in minutes per year or as a percentage of the time the services are provided (Rahul Ghosh et al. (2014)).

Hitherto fault tolerance practices do not provide adequate solutions in the clouds. Any attempt to develop suitable applications falls into the gap due to the lack of knowledge of the infrastructure by the user. The most common practice is the replication of VMs and their migration to other servers.

To examine the availability of clouds, we must first look at the infrastructure and the risks that may cause service disruption.

On the base there are servers with the appropriate resources. On each server the hypervisor using the real resources makes the virtual ones to offer them to VMs. The minimum used for their interconnection is a 1Gbps network card connected to a switch. Each switch is connected to 2 larger (primary and backup) aggregation or distribution switches, thus locally forming a cluster. Aggregation switches are connected to aggregation routers to form the data centers. Data centers are joined by core routers and thus the cloud is created (Ravi Jhavar and Vincenzo Piuri (2012)).

Each part of the cloud is a possible cause of problems. Servers may have problems with the hardware or even with the hypervisor and hence with VMs. The network can also cause trouble with a switch, router, or even a physical interface. For this reason, the user must make copies of the applications in different zones of the cloud. Electricity is also a potential source of problems. There must be uninterruptible power supply mechanisms (UPSs, generators), and each cluster must have power by another line of electricity.

The biggest possible problems may arise from cloud management errors, but they are difficult to test and when they emerge they may be radical so they are only statistically examined.

At the application level copying and migration can be done in three ways. The first is semi-active replication. In this approach, the data are imported either to all copies or to the master VM first and at regular intervals they are transferred to the rest. All copies process the data but only the master sends the results to the user. If the master VM fails immediately a copy takes over and creates a new one to take its place. So the availability is very high but we have great resources and energy consumption. The second way is semi-passive replication. Here, at regular intervals, the master VM status is checked and transferred to copies that do not run the application. In case of failure of the main machine, a backup is started that knows the last state stored in it. The data that have occurred till the occurrence of the problem are lost. In this way the availability is smaller but fewer resources are consumed since the copies are not active. The third and last way is passive replication. At regular intervals a backup of the application is taken, and if a problem occurs, a new VM is being recovered. Some data are lost and availability is the smallest, but also the resources consumed are the fewest.

The installation of virtual machines and copies can be done in 3 locations. The first case is to be done in the same cluster on the same local network (local area network, lan). Here we have insignificant latency and high bandwidth for communication between VMs and fast change of hardware when there is a problem, but availability is limited, since cluster problems (network, current etc.) will affect all VMs. The second case the VMs to be in different clusters but in the same data center. Thus, we have a good latency and bandwidth, but of course worse than the first case and better availability, which is unfortunately limited by problems that affect the whole data center. In the third case, VMs are placed in different data centers, which can be in different cities, countries or even continents. We have a lot of latency and low bandwidth but the highest availability (Ravi Jhawar and Vincenzo Piuri (2012)).

The physical servers used in the clusters can be active and appropriately configured with the VMs for the services (hot servers) providing the highest availability, active but not properly configured, reducing energy consumption but at the same time also the availability and, finally, inactive. Inactive should be put into operation and configured for use. The availability they offer is the smallest, the reduction in energy consumption, of course, is the highest. When there is a problem on

a hot server, a warm is used (or a cold if there is not a warm). After the repair, the servers are repositioned in their places so that their number and the availability that this entails are not reduced (Rahul Ghosh et al. (2014)).

3.5 Vendor lock in

From the very first moment cloud computing was seen not only as a promising technology but as the "future" of IT. Nobody could or wanted, to predict which sectors it could cover and up to what point it could reach. It was thus left to be developed both by the academic community and by the commercial companies that emerged on the path, without strict control, based on some old and thus incomplete standards. Companies, trying to differentiate themselves and offer things that the competition did not have, developed new standards that have been established as de facto standards due to their penetration into the market. But this has led to incompatibility and lack of interoperability between the various cloud implementations and finally to the entrapment of customers, something that is not desirable for themselves but desirable for the companies.

This entrapment of the customer in a company is along with security the most important reasons some people fear and ultimately avoid cloud services. Due to the incompatibility it is very difficult even to transfer the applications and the data of the client to another provider (portability). Besides the fact that in critical applications it is necessary to carry out any transfer if possible without downtime. Of course, customers are afraid they may even lose their data if, for example, the company closes and can't maintain its infrastructure. The ideal scenario for customers is to be able to choose services according to their quality and availability, as well as cost. And eventually cooperate with any provider they feel right at any time. To be able to do this, it must be possible, in addition to the transferability and interoperability between the clouds, as well as their consolidation. And providers, of course, have to want interoperability and consolidation, since they can offer power, capacity or other services from another partner when they can't for any reason themselves. The possibility of horizontal consolidation is something that is being promoted today and it is being attempted to be imposed politically by organizations such as the European Commission.

To make this possible, common standards must exist and be used. The most popular are open standards, because the standards established by the companies, even though they have proved their value commercially, depend on their development by them. Even open standards that are based on commercial ones, risk compromising their compatibility with them if the company makes a decision to diversify it, trying to keep its clients trapped.

In terms of management, APIs should be HTTPS-based and RESTful, such as the Open Grid Forum's (OGF's) Open Cloud Compute Interface (OCGF) and the Distributed Management Task Force 's (DMTF's) Cloud Infrastructure Management Interface (CIMI). Organization for the Advancement of Structured Information Standards's (OASIS's) Topology and Orchestration Specification for Cloud Applications (TOSCA) is useful because it describes the relationships and operational management of services and applications on the cloud, while the Storage Networking Industry Association's (SNIA's) Cloud Data Management Interface (CDMI) describes how the various functions for the data in the storage media will be made.

DMTF's Open Virtualization Format (OVF) is an open standard for virtual machines that enables them to be transferred from one cloud to another.

Unfortunately, most clouds use their own user authentication and authorization techniques. An existing standard for authentication in grid computing is the X.509 Public Key Infrastructure through the Grid Security Infrastructure (GSI). On this the OASIS Security Assertion Markup Language (SAML) is based which is used by Shibboleth. Shibboleth integrates authentication and authorization schemes for different clouds.

The presentation and extraction of the information of the various capabilities of the consolidated clouds is very important. The Grid Laboratory Uniform Environment (GLUE) Scheme of the OGF helps to present this information and create a cloud profile.

Finally another problem in the consolidated clouds is charge, since each cloud has its own methods and different costs. OGF's Usage Record (UR) 2.0 provides a unified way to share and exchange these data (Álvaro López García et al. (2016)).

4 Investigated IaaS platforms

4.1 Amazon EC2

Amazon Elastic Compute Cloud (EC2) is a web service that provides computing infrastructure to its customers. It is possible to be customized according to their requirements.

Through a simple interface, it enables users to acquire computing resources quickly and easily. Users have the ability to increase or decrease the resources they rent according to the needs of their applications at any time and to pay accordingly. They do not have to rent the resources they will probably need from the start.

The Amazon EC2 virtual machines (VMs) are called Instances. The user can create one from the beginning with the power, the operating system, the applications and all the resources he wants, or use one of the ready ones to use templates called Amazon Machine Images (AMIs) if one can cover his requirements, thus gaining time. He can use the VM's own storage called Amazon EC2 Instance Store or a virtual storage from the Amazon Elastic Block Store (Amazon EBS). In the first case the data are lost when the VM goes out, while in the second the storage space belongs to the user since he pays it and the data are not lost. The data of the user's sites can also be stored on the Amazon Simple Storage Service (Amazon S3).

EBS offers storage to the user in the format he is accustomed to on the computers he owns. The data are still present in the EBS volume regardless of whether the instances are in running state or not. However, each volume can be assigned to only one instance. Instead, in one instance we can assign many volumes. There are 2 storage categories, those supported by Solid State Drives (SSD) drives and those supported by HDD (Hard Disk Drives) drives. Apparently if the application requires high input / output data rates, the first category will be preferred, whereas if large volume and / or continuous data transfer is required, the second category should be preferred. Each category has 2 subcategories with different performance and charge.

The user can take snapshots of his storage spaces and share them with others. There are also public snapshots that offer ready to use computing systems and can be used by anyone to make his own ones according to his needs.

Encryption is used to the storage and the snapshots for security reasons.

Amazon Elastic File Storage (EFS) is a Linux-based file system that can be assigned to any instance or even to a user's computer outside of EC2.

The Amazon S3 is a reliable and inexpensive form of storage accessible either from EC2 or the web. It can be used to store copies of user data and applications. EC2 uses it to store snapshots and backups of instances. Its speed is lower than that of previous formats.

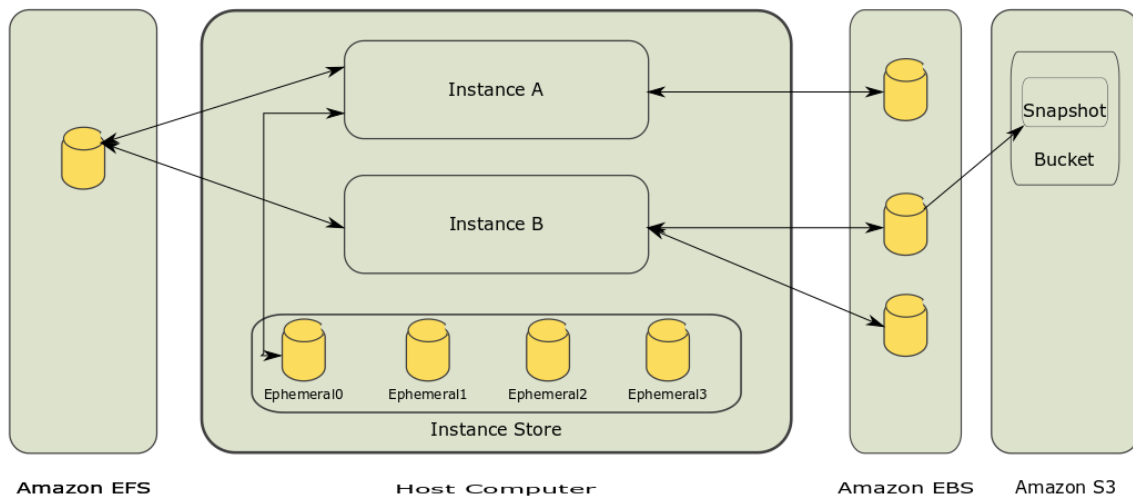


Image 6: The relationship between different types of storage.

<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/Storage.html>

For rarely used data, archiving and back-up for many years, even decades, Amazon offers Glacier, a very cheap and reliable form of storage. It's clearly cheaper and slower than S3.

<https://aws.amazon.com/documentation/glacier/>

Users have also access to VMs and through DNS names.

Initially, users can have up to 20 reserved VMs (Reserved Instances) and an equal or smaller number of on demand VMs (On Demand Instances), depending on

their size and power. If they need more, they should ask for it. They are also entitled to a number of Spot Instances depending on the geographical area.

On demand VMs can be initiated and terminated at any time by the user, with the appropriate charge. The Reserved, are reserved for him for a period of time, 1-3 years. Due to the long-term commitment, there is a great discount on rent. In Convertible Reserved Instances, the discount is a little smaller but there is the ability to change the machine configuration during the lease. For Spot Instances the user gives a higher value. As long as this value is above the average given by the others, the VM works, otherwise it doesn't. EC2 also provides physical servers, dedicated to a user, with anything this entails.

There are also Cluster Instances. It is possible to join them over a network for High Performance Computing and demanding web applications.

Currently, for instances, the following operating systems are supported: Amazon Linux, Ubuntu, Windows Server, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Fedora, Debian, CentOS, Gentoo Linux, Oracle Linux and FreeBSD.

There is a limit to the number of emails that can be sent from an instance. If more are needed, a request must be made again.

The charge for using instances is per second, but the account is shown per hour with decimal subdivisions. Charging starts from the moment the activation command is given until the VM operation ends. The charge for the transfer of data between different regions is made according to the cost of the telecommunication providers. Both the instance that sends data and the one that receives are charged. In general, both data input and output are charged.

The provided machines are divided into 5 categories. General Purpose, Computational Power, Increased Memory, Increased Storage Space, and Accelerated Computational Power. The General Purpose is for common applications. T2 stands out, which has a variable processor power. Power boost is used for short periods of time, as the Engine needs to collect extra power when it is down. Computational Power is for increased requirements such as High Performance Computing. Accelerated Computational Power is for 3D graphics, high performance computing and Artificial Intelligence.

The user controls which machine groups will communicate with which others. The same can be done for IP address groups. He also controls which groups will communicate with the Internet.

Each VM has a private and a public IP address for its communication. 5 elastic IP addresses are initially given per user unless requested more. These addresses do not change as long as they are reserved. They are charged and used where a static address is required.

Physical computers are installed into different Regions which are divided into Availability Zones. The user has access only to the resources of his Region. Each Availability Zone is cut off from the rest, so a disaster in a Zone, natural or not, can't affect others.

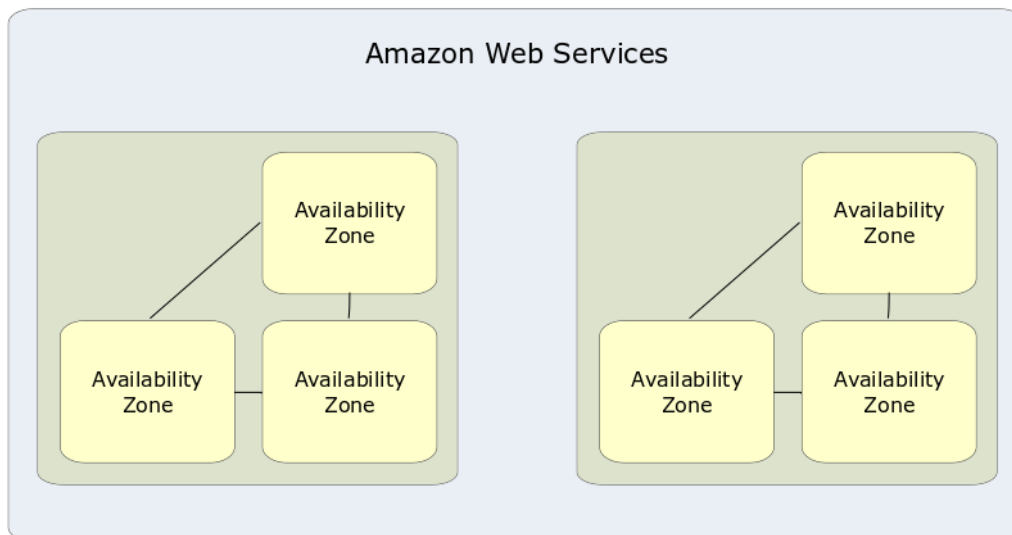


Image 7: Availability zones and regions.

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>

Until now, EC2 has been using a Xen-based hypervisor. Now it has developed its own, the Nitro Hypervisor based on core Linux Kernel-based Virtual Machine (KVM) technology. It offers processor and memory isolation of VMs, while network and storage are implemented with separate hardware rather than software. This also leads to better performance. Also, storage is built up by Solid State Drives (SSDs) that offer much faster speeds.

EC2 supports 2 types of virtual network cards. The first is Enhanced Networking. It uses the Single Root I / O Virtualization method that offers better input-output performance and lower power consumption. Most types of instances offer it. But it is only used in Amazon Virtual Private Cloud (VPC). The second is based on the Intel® 82599g Virtual Function Interface. This can also work outside the Amazon VPC.

Monitoring the user's infrastructure at EC2 is done through Amazon CloudWatch.

The user can use Auto Scaling to increase or decrease the capacity of its infrastructure automatically under certain conditions he defines. He can specify a group of VMs that will start or end in accordance with the conditions set. The maximum number of VMs allowed to a user can't be exceeded.

Applications can also benefit from Elastic Load Balancing either through the Classic Load Balancer that routes data traffic according to information from the network or the application between instances or through the Application Load Balancer that takes into account and information of the call content of the application.

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html>

The user has the ability to import a VM from his own data center and thus create an instance. He also has the ability to export an instance to its own infrastructure. VMware ESX VMDK, VMWare ESX OVA, Microsoft Hyper-V VHD, and Citrix Xen VHD are supported. The VM to be exported must be terminated. Can't be exported an instance that has more than one network card or EBS storage. Only the boot disk with the operating system can be exported.

<https://aws.amazon.com/ec2/faqs/>

<https://aws.amazon.com/documentation/ec2/>

<https://aws.amazon.com/documentation/>

Energy saving leads to lower operating costs. This is achieved primarily by making more effective use of available resources and better data center design and operation. By shifting to AWS cloud, customers use on average 77% fewer servers, 84% less power and 28% cleaner energy for a 88% reduction in CO2 emissions. In a large cloud provider, server resources are utilized at 65%, while at customer facilities

they are utilized at 15%. This means that in the cloud the applications require 23% of the resources, so saving is 77%. The power consumed by a data center is lost to a percentage in distribution, cooling, lighting and other causes. The extra power lost on the client's premises compared to the one consumed by the resources is 70% when in the cloud only 20%. That means saving 29%. So the 23% of the servers is necessary that consume 71%. This means that 16% of the power is needed, which makes a saving of 84%. The average of g CO₂ per KWh of energy in June 2015 was 545. In AWS it was 393. 28% less. With 16% less energy that emits the 72%, we have 12% pollutants, so 88% less. By using renewable energy the target is 100%.

<https://aws.amazon.com/blogs/aws/cloud-computing-server-utilization-the-environment/>

<https://aws.amazon.com/about-aws/sustainability/>

<https://www.amazon.com/p/feature/gkkwdp34z5ou7ug>

There are many services and tools to ensure the integrity, confidentiality and availability of systems and customer data. The platform uses autonomous and multilevel infrastructure controls seamlessly in both the data centers and the network. Amazon is responsible for infrastructure security, but the customer is also responsible for his applications and data that should not cause problems to others. The level of access to customer data depends on their own implementation.

For network security, firewalls are provided to create private networks with controlled access, while encryption is performed when data are transferred with TLS to all services. Private, dedicated connections to customer facilities are provided. So are technologies that restrict Distributed Denial Of Service (DDoS) attacks.

For the data that are stored, encryption capabilities are available in all storage formats. The client can choose whether AWS will manage the encryption keys or himself while hardware-based key storage is provided.

APIs allow the implementation of encryption and data protection in any service.

Access control policies can be applied everywhere. There are capabilities of identity and access management with separate user accounts with different levels of

authorization, multifactor authentication even with hardware-based devices and integration with existing directories at the customer's premises to reduce the load of the administrators and improve the quality of the user experience.

The tracking and logging tools give all the details of the calls to the various APIs and alert when something unusual happens.

AWS Trusted Advisor is an on-line tool serving as a specialist advising on best cloud implementation practices for better resource management, greater security and money saving. AWS Account Teams are the first contact for security issues while AWS Enterprise Support has a 15-minute response time and is always available.

The AWS Professional Services and AWS Partner Network helps customers implement the best security policies in accordance with proven practices and to be consistent with the requirements of compliance with national and international standards and legislations.

Of course, European legislation on the protection of personal data is followed entirely. Conformity checks are continuous across all data centers with tools such as AWS Config and AWS CloudTrail, relieving customers of the relative load and anxiety.

https://d1.awsstatic.com/whitepapers/Security/Intro_to_AWS_Security.pdf

<http://d0.awsstatic.com/whitepapers/Security/AWS%20Security%20Whitepaper.pdf>

https://d1.awsstatic.com/whitepapers/compliance/AWS_Risk_and_Compliance_Whitepaper.pdf

4.2 Google Compute Engine

Google Compute Engine (GCE) virtual machines are called Instances (Virtual Machine Instances). The user can create an instance by using a Public Image of an operating system that he wishes, a Custom Image from the beginning or by inserting a ready system into the Google infrastructure. Public Images currently support the following operating systems: CentOS, Container-Optimized OS from Google, CoreOS, Debian, Red Hat Enterprise Linux (RHEL), SUSE Enterprise Linux Server (SLES), SLES for SAP (version of SUSE for the ERP SAP), Ubuntu, Windows Server and Windows Server with built-in Microsoft SQL Server).

GCE offers predefined types of VMs but also the ability to the user to create an instance as he wants. First, there is the Standard type for the most common applications. The second type is the one with more memory (High-memory) and the third is the high-CPU, oriented to CPU power. Last of the standard types is Mega-Memory, for applications with high demands both in power and memory. Up to 16 discs with 64 TB capacity can be assigned to all of these types.

There are also shared-core types that run on a processor thread for a short period of time, taking advantage of the idle periods of the processors. They are much cheaper and more suitable for non-demanding applications. There is the possibility to temporarily increase power over short periods of time (and only in this type). Up to 4 discs with a capacity of 3 TB can be assigned.

Graphical Processing Units (GPUs) can be added to each type except the shared core. However, many GPUs can only be added to types with a lot of power and memory.

The Preemptible VM Instances may be useful in some applications. They are provided by GCE when resources are available, but may be terminated by the provider whenever needed by another user and another application. There is no SLA. They are useful as extra power in applications that will need it in addition to the normal application support.

There are 4 types of storage space in GCE. They are the Persistent disks that can be normal hard disks or SSDs, local SSDs and Cloud Storage buckets. Additionally, the user can create an independent file server and RAM disk for better performance.

Persistent disks are network disks that each instance uses as local. They are independent of instances and do not lose their data if the instance is terminated. Data security and performance are the responsibility of the provider. Their size increases according to the data. Persistent disks based on SSDs have significantly better performance than ordinary ones. Data are being encrypted and snapshots can be created for security.

GCE offers the Virtual Private Cloud (VPC) network. Each client creates a Google Cloud Platform (GCP) Console Project that can have up to 5 VPCs. Each network has subnets that have a range of IP addresses that they provide to instances. Each instance has an internal IP address. Each network has an initial configuration and its own firewall rules. The client can configure a network from the beginning or continue with the original configuration.

Each instance has an interface but it can get even more if needed. Only IP v4 is used. The initial configuration of the firewall prevents any data traffic to and from instances. The client should create rules that allow specific traffic. Each rule is in two directions, inward and outward. GCE prevents Internet traffic on some ports like e-mail.

An instance that will communicate with networks outside of its VPC should have an external IP. Traffic within the VPC can be done via the internal. The VPC network comes with predefined rules for routing, but the client can modify them. VPC provides 2Gbps bandwidth per vCPU (per core) with a maximum of 16 per instance. This is theoretically the maximum within the VPC with internal IP addresses. With external bandwidth decreases due to other routes. Persistent disks consume bandwidth because they are network disks.

Both internal and external IP addresses are distinguished in static and ephemeral. The ephemeral ones remain as long as the resource (instance, rule etc.) remains active. If it is shut down or restarted, the address is lost and another address is specified. For applications that the IP is important for routing traffic as long as the project lasts, there are static IPs. They belong to the customer and remain in the resource until otherwise specified. A name can also be given to Instances that can help data traffic if IPs are ephemeral.

GCP offers load balancing in terms of data traffic. So we can have application scaling and share the traffic to many VMs. The service is offered by GCP, so it has high

availability. The status of VMs is monitored to let load balancing avoid the ones with problems.

GCE also offers automatic scaling (and de-scaling) –named autoscaling- if we have created a group of instances. Instances must be identical to have autoscaling. Then, the number of VMs serving an application may increase if the requirements increase and decrease when the requirements are reduced accordingly. For this to happen, one - and only one- policy must be assigned and followed.

The policy may refer to the use of processing power, with upper and lower limits leading to adding or removing VMs or to load balancing data. For automatic scaling, the administrator can use other VMs status measurements given by the platform or set his own.

The resources of GCE exist and are distributed in regions and zones. Today in North America there are 4 regions, in South America 1, in Europe 3, in Asia 4 and in Australia 1. Each region contains a number of zones. There are resources that are shared by a zone and must be from the same zone to be combined like instances and persistent disks and resources that are regional or global like VMs images.

Each zone is independent of any other, it has its own data center and an infrastructure failure only affects the particular zone. The client has to design his infrastructure so that his machines are in the same zone or area for low latency in the network, in an area close to his client's networks and at the same time to distribute copies of his infrastructure to other zones or areas to enjoy high availability in case of major damage. The cost of communication between zones and areas varies.

All GCP projects upon creation have a single user, the creator of the project. He can then give access to other users, with different rights if needed. There are many roles that can be assigned to a user, network administrator, instance administrator, security administrator etc. There are also the roles with only monitoring rights, not intervention.

If the creator of the project wants to give access to a user on a specific or all of the VMs via SSH, he can add the SSH public key of the user to the specific VMs or the project without giving him extra rights. If there are applications in the project that need access to other services and APIs, the project creator can use the service accounts that

automatically give access without requiring any other user with a role. The service account has an identity used for authentication and authorization.

For the import of existing servers from the client infrastructure or from another cloud provider, GCE offers the VM Migration service. Windows Servers from version 2008 and later and the most common Linux distributions are supported. For export, GCE images use RAW format. Then other tools, the other provider's or independent manufacturers' like Oracle Virtual Box, must be used to convert to other formats.

The charge is made per second, once a VM starts until the end of its operation. The minimum charge is one minute. There are discounts for instances that are used more than 25% of a month and much greater ones if a longer contract (one or three years) is signed.

<https://cloud.google.com/compute/docs/>

In addition to autoscaling, GCE offers the Rightsizing Recommendations service that is still in beta. They are machine type recommendations that are designed to help the user optimize the use of VMs resources. They are automatically generated according to system metrics collected by Google Stackdriver Monitoring. The user can take advantage of these recommendations to redefine the configuration of VMs and make better use of their resources. The metrics are collected every 60 seconds averagely and involve high or low power and memory load.

A specific, real type must be used to reformat a VM. The cost of a VM is calculated based on last week's usage after being converted to monthly. It then compares to the cost of the proposed one. Stackdriver offers a Monitoring Agent that collects CPU, disk and network readings to automatically help with recommendations. The Agent must be installed on the VM.

<https://cloud.google.com/compute/docs/instances/apply-sizing-recommendations-for-instances>

Google aimed for 2017 to be based on 100% renewable energy, mainly wind and solar. Consequently, CO2 emissions reach 0. The extra power consumed in a data center, other than that of resources (cooling etc.), is only 12%.

<https://cloud.google.com/environment/>

36% of the servers and 22% of materials are being rebuilt. Due to the efficiency of data centers, 50% less power is needed. The infrastructure is ISO 50001 certified.

Security on the Google platform is controlled and implemented by the Information Security Team. This team maintains defense systems, makes controls, builds the infrastructure, and implements security policies. Google has an SSL default application policy.

<https://cloud.google.com/security/>

Data center security is multi-layered and includes specially designed by the company electronic access cards, bars for vehicles, fences, metal detectors and biometric systems. The floor inside has intrusion detection laser. There is a camera monitoring system and the logs are dealt with when an issue arises. Security patrols are made by a security service, while less than 1% of the employees need access to the data centers.

The infrastructure is based on thousands of identical, custom-built servers. All the hardware, the networking and the Linux operating systems are implemented safety-oriented. To protect even the boot process, Google has developed a dedicated security chip called Titan that allows for hardware and software verification to provide a strong hardware-based identity.

<https://cloudplatform.googleblog.com/2017/08/Titan-in-depth-security-in-plaintext.html>

There are many controls to protect customer data. Access to storage is done through authentication and authorization, while engineers' access is also controlled. This is achieved by a role and group management system that uses a security protocol and authenticates engineers with short-term personal public key certificates.

Hard disks that stop being used by a customer are subject to a data destruction process. Authorized personnel make the destruction, which is re-checked while the results are logged. They can then be reused unless they have a problem with the material. In that case they are stored in a safe place until they are physically destroyed.

All services are provided through a secure infrastructure of a global API portal. Encrypted SSL / TLS channels are used, and each call must include a short-term authentication token generated either by user-entered codes or a private key.

<https://cloud.google.com/security/encryption-in-transit/>

Customer data are automatically encrypted while stored. There are many encryption mechanisms. Persistent disks use 256-bit Advanced Encryption Standard (AES-256) and each key is re-encrypted by a group of master keys.

<https://cloud.google.com/security/encryption-at-rest/default-encryption/>

Google's global network helps secure data in traffic by reducing intermediate nodes. Cloud Interconnect and VPN allow encrypted channels to be created between customer premises and Google, skipping the public Internet.

Intrusion detection mechanisms are provided where there is data input.

Google's platform and infrastructure are certified for the best-known compliance standards and allow for independent security and privacy checks.

<https://cloud.google.com/security/compliance>

4.3 Microsoft Azure

Microsoft Azure virtual machines are called Virtual Machines. The client can create his own VMs from boot images or import them from another infrastructure. He has access to Windows VMs via the corresponding Remote Management service while to Linux VMs via SSH. But he has to import his own SSH keys. There are predefined types to choose from, depending on power, memory, storage and the interface he wants.

First we have the General Purpose type, which is the basic for the most common applications. Then we have 3 types, for high processing power, plenty of memory and high speed storage respectively. Finally we have the graphics-capable type with one or more high performance graphics cards and the highest-performance type with the highest power, memory, faster and larger storage and special high performance network cards. These are mainly used in parallel processing applications when very high speed and low latency in communication between machines are required. The latest type has 32 CPU cores, 448 GB RAM, and 6.5 TB local SSD drives.

The images used can only be bootable, with the operating system that may be the latest versions of Windows Server and the most popular Linux distributions, or integrated images along with data drives. They may come from Microsoft, a partner or the Azure community, tailored to specific needs.

Azure offers the Batch, a batch processing service. Batch also uses low priority VMs, which are temporary and result from resources that are not fully utilized. They are cheaper, but they can return to the application they belong to without approval. They cover temporary high power needs for parallel processing helping as long as they exist. VMs can be further configured after the initial creation with Azure VM Extensions.

<https://azure.microsoft.com/en-us/services/virtual-machines/>

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/>

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-specialized>

<http://www.dotnetcurry.com/windows-azure/1299/microsoft-azure-platform-services-overview>

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes>

<https://cloud.google.com/docs/compare/azure/compute>

For scaling there is the Azure Autoscale. It can specify a minimum (for the continuous operation of the application) or even a maximum (to avoid exceeding the budget) number of virtual machines or add and remove machines according to policies defined by the user. A set of identical machines is created called VM Scale Set. VMs are derived from a Resource Manager template. Portals, Powershell, CLI, REST APIs and various Software Development Kits can be used.

In Azure there are 2 types of automatic scaling, scheduled and dynamic. In scheduled, machines are added and removed at a predetermined time. In dynamic, the addition and the subtraction of the machines is done according to thresholds (upper and lower) of measurements concerning the use of computational power, memory, network etc., or specific output data. This is the most common horizontal scaling. But there is also the vertical. In vertical, the VM type can be changed, according to the available types in a region. Power, memory, etc. may be increased or decreased. However, this is done manually and the machine must be restarted.

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/autoscale>

<https://cloud.google.com/docs/compare/azure/compute>

Microsoft for managing VMs uses an Azure-specific version of Hyper-V, the hypervisor it uses on Windows Server. Consequently, VHD format must be used to import images from another infrastructure. The same applies to export.

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-specialized>

Azure offers isolated virtual networks called VNet. These are region resources. In an infrastructure we can have many such VNets and divide them into subnets. VMs created in a VNet automatically and without additional configuration communicate with each other. The IP address range of a VNet must be set in advance. When they are created, VMs have an internal IP address that is used within VNet. They can also have an external for internet connection. They are dynamic addresses, so they are lost if the VM is terminated. For an IP to be permanent, if the application requires it, a static IP

address must be set. Static addresses are automatically granted, not selected by the administrator. A VM can have as many as 32 vNICs.

A Network Security Group (NSG) can be defined to manage traffic that is permitted or prohibited from and to a VNet. It is a set of rules for this purpose and may involve a subnet or even a single network card (vNIC). If it is set to a vNIC it only applies to this, if it is set to a subnet it applies to all VMs belonging to it. The rules apply to both directions, they may apply to any port and have priority. High priority is considered first.

Administration is done through the CLI.

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/tutorial-virtual-network>

Azure also supports stateless rules with Access Control Lists (ACLs) at the VM level.

<https://cloud.google.com/docs/compare/azure/networking>

For VPN (Virtual Private Networking) services, Azure offers the VPN Gateway. It can create a tunnel between an external network and a VNet and ensure a secure connection. It also has a routing service.

Azure's resources are distributed across many geographical areas around the world. They are called regions. There are many data centers in each area. Until November 2017, there were 36 regions, while another six were announced. In America there are 17, in Asia 13 and two more have been announced and Europe 6 with another 2 being announced. Also 2 have been announced in Africa. In China, there is a collaboration with 21Vianet, one of the largest internet providers in the country and in Germany, with Deutsche Telekom's T-Systems International GmbH, with the aim of keeping customers' data in the country.

Not all products are available in all regions.

<https://azure.microsoft.com/en-us/regions/>

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/overview?toc=%2fazure%2fvirtual-machines%2flinux%2fclassic%2ftoc.json>

Microsoft has announced a Service Level Agreement for a VM of 99.9% if it is implemented with premium storage for all drives. To reach 99.95%, at least 2 VMs

must be implemented in an availability set. This means they have to be in different data centers with a different maintenance period.

There are 4 types of storage, Blob storage, File storage, Queue storage and Table storage. To use them, the user must first create a storage account.

Blobs are files similar to the ones of regular computers. They are stored in containers that are the equivalent of the folders. There are 3 kinds of blobs. Block blobs store files sized as 4.7TB. In page blobs random access files are stored sized as 8TB. This is where the VM VHDs are stored. Append blobs store app attachments such as loggings, possibly from multiple VMs at the same time. There is access through Internet (URLs), REST interface and the Azure Software Development Kit. If the user wants to store very large files he can send his drives to Microsoft and copy them locally to avoid the telecom cost.

There are 3 types of blob storage, hot, cool and archive. The first is for data we use often. It has the fastest access and the best availability. Cool is sacrificing little speed and availability for lower costs. It is for data that will be stored for at least 30 days. The archive is off line for data stored for at least 180 days. Its cost is the 10% of the one of the hot blob storage.

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers>

File storage is implemented with Azure Files. It is a file sharing service with the Server Message Block (SMB) protocol. This way many VMs can share the same files with read and write permissions. Files are also accessible through URLs (including shared access signature (SAS) tokens), REST interface and the Azure Software Development Kit. Storage accounts use authentication to provide access.

Queue storage is provided with the Queue service. Used to store and retrieve messages as large as 64Kb. Messages may be for communication e.g. between applications and the platform (to do something).

Table storage is a service for storing large amounts of structured NoSQL schemaless data in the cloud. The tables increase as the need grows.

<https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>

Storage for VMs is of 2 types. There is the Premium Storage based on SSDs and the Standard Storage based on HDDs. Standard Storage supports all types of

storage, while Premium only supports page blobs. For demanding applications, Microsoft recommends using Premium Storage. By integrating multiple drives, we can achieve 256TB capacity, 80,000 IOPS (Input / Output operations per second) and 2,000 MBps (megabytes per second) per VM.

Two types of disks, Unmanaged and Managed are provided. On Unmanaged Disks the user has the management and responsibility to store VHD files in different storage accounts. They are stored as page blobs. On Managed disks Azure has the management of the storage accounts. The user simply determines whether it is Premium or Standard and the size. Microsoft recommends using Managed Disks.

Azure provides many ways to efficiently manage resources achieving cost reduction. One is the Azure Policy. The client can create, apply, and manage policies, that is, resource management rules according to the needs and goals of the company / organization and the SLAs. From the beginning, Active rule categories refer to permissible and non-permissible types of resources, VMs, locations, and storage accounts.

<https://docs.microsoft.com/en-us/azure/azure-policy/azure-policy-introduction>

Another one is Azure Cost Management. With this, the client monitors, applies and optimizes the resources used as well as their cost. With this service the user monitors the efficiency of VMs and can control their right-sizing and the VMs that are not used.

<https://azure.microsoft.com/en-us/services/cost-management/>

<https://azure.microsoft.com/en-us/blog/new-azure-management-and-cost-savings-capabilities/>

Advisor provides recommendations for optimizing resources for high availability, security, performance and low cost. It also gives guidance on the implementation of the advices.

<https://azure.microsoft.com/en-us/services/advisor/>

<https://docs.microsoft.com/en-us/azure/advisor/advisor-overview>

Azure Reserved VM Instances (RIs) can be reserved for 1-3 years with a discount of up to 72% or 82% if a Windows Server license with software assurance that the customer already has is used.

<https://azure.microsoft.com/en-us/blog/new-azure-management-and-cost-savings-capabilities/>

VMs are charged per minute, but extra seconds are not charged.

<https://azure.microsoft.com/en-us/blog/achieve-better-savings-with-best-in-class-cost-management-on-azure/>

Microsoft aims to reduce CO2 emissions gradually. By mid-2016 only 44% of the data centers were based on renewable energy sources. The target is to reach 50% at the end of 2018 and 60% at the beginning of the new decade. Meanwhile, the company is buying renewable energy certificates (a form of fees) reaching 100% from 2014.

<https://blogs.microsoft.com/on-the-issues/2016/05/19/greener-datacenters-brighter-future-microsofts-commitment-renewable-energy/>

Azure puts emphasis on security in every sector. Security concerns both services and data and the infrastructure that supports them.

Identity and Access Management is done by Azure Active Directory (AD) and Azure Storage. AD is a warehouse and a mechanism for authentication, authorization, and access control for users, groups, and objects. It uses protocols such as SAML 2.0, WS-Federation, and OpenID Connect while RESTfull graphical APIs with OAuth 2.0 support can deploy applications that will utilize it. It can coexist with the client's Active Directory with the same credentials. Authentication is also provided in Storage.

The Multi-Factor Authentication service offers mobile verification while Domain Services access to a domain without another domain controller. Finally, with Active Directory B2C, users can grant access with social networking identities.

Azure functions are based on the principles of Separation of Tasks and Minimum Privileges. In data centers, support staff only has access to the client's permission, the actions are recorded and the license is revoked at the end of the incident.

Data protection applies both during transport and storage. Encryption is for data, drives, files, communications, applications, and services. Encryption can take place before the data are transferred to Azure and the keys can be located on the client's premises. BitLocker Drive Encryption can be applied to VHD files. Table-level and

column-level encryption (TDE / CLE) is supported in SQL Server virtual machines. Various encryption mechanisms such as SSL / TLS, IPsec, and AES are supported.

The hypervisor of VMs offers complete isolation from physical servers and thus between VMs themselves, enhancing security. By default, all data traffic is banned in a VM until other rules are applied. DHCP and DNS traffic and the public Internet are excluded. Nobody can have network access to the physical infrastructure.

Access control to the network is done by isolating VLANs, Access Control Lists, Load Balancers, and IP filters. There is complete control of outgoing and incoming traffic, protocols and ports.

Site-to-site VPN and point-to-site VPN are provided to enable communication between client's data center and the cloud. Connections require authentication with SSL / TLS. Secure management certificates and security protocols such as SSTP and IPsec are supported.

ExpressRoute allows secure connections (lines) between client premises and Azure data centers. They are not through the public Internet and provide greater reliability, security and speed; many times and considerably less cost.

Full security event logging and monitoring is offered, such as changes to IP addresses from DHCP and DNS, attempt to access ports, protocols and addresses prohibited by network design, security policy and firewall changes, user account creation and device drivers installation.

There are the Management Agents and the Azure Security Monitor agents that monitor every resource, physical or virtual. These agents are not installed on customers' VMs and do not give them extra load. Logs are collected from networked devices using the Syslog protocol and from servers with Microsoft Audit Collection Services (ACS).

<https://docs.microsoft.com/en-us/azure/security/azure-security-getting-started>

4.4 OpenStack

OpenStack is the most widespread platform for creating IaaS. It is open source software and distributed free of charge. It controls and coordinates computing power, storage, network, etc. to create IaaS. OpenStack Foundation, a nonprofit organization coordinates the effort of its development and dissemination.

The cloud computing goes one level further of virtualization. In virtualization, the various resources are unified, homogenized (as they can come from different manufacturers with different technologies, drivers, etc.), separated and distributed by the hypervisor. OpenStack provides a group of APIs that lead these resources to a higher level of abstraction by turning them into tanks that the various cloud computing tools use. These tools create the cloud environment according to the NIST definition.

<https://www.redhat.com/en/topics/openstack#>

<https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp>

OpenStack consists of many projects, 6 of which are fundamental and have to do with computer systems, network, storage, identification and images. They are Nova, Neutron, Swift, Cinder, Keystone and Glance.

Nova (OpenStack Compute) is the platform's main tool. It provides on demand access to computing resources managing virtual machine networks. These resources can be "conventional" VMs, containers, and even physical servers (bare metal). It works with virtualization technologies such as KVM, VMware, Xen, Hyper-V and Linux containers such as LXC and LXD. The collaboration with physical servers is through OpenStack Ironic, the platform's virtualization technology. Thus it can collaborate with and utilize almost any existing computing system. At present, not all features in all technologies are implemented and available. More has been done for the KVM, and a very good implementation has been done for Xen.

<https://docs.openstack.org/nova/latest/user/feature-classification.html>

Nova provides horizontal elasticity. It works directly with Glance that provides images, Neutron for network and Keystone for identification. It is handled either

through Horizon, which is the Web-based interface or through the OpenStack Client, which is the command line interface (CLI).

<https://www.webopedia.com/TERM/O/openstack-nova.html>

<https://github.com/openstack/nova>

<https://docs.openstack.org/nova/latest/>

Neutron (OpenStack Networking) offers the services that are needed for on demand, scalable and independent from technology and manufacturers network abstraction. It provides "Networking as a Service" by merging virtual network devices (vNICs) into a virtual network. It manages IP addresses by providing static or dynamic ones, as required. At the same time, there are the "floating" IP addresses that allow the dynamic routing of data traffic between infrastructure resources in the event of a failure.

Neutron provides capabilities for building complex network topologies and implementing advanced policies such as those needed in web applications. There are also plugins that offer advanced features such as Layer2 to Layer3 tunneling for VLANs and Quality of Service (QoS) guarantees. So anyone can have services like LB-aaS (Load Balancer), VPN-aaS, firewall-aaS, IDS-aaS (Intrusion Detection System), data-center-interconnect-aaS.

It is managed through the Horizon UI and allows VMs to start on specific networks while it can connect its own Layer 2 networks to physical VLANs through API extensions. It can support Software-Defined Networking (SDN) technologies through the OpenFlow protocol, enabling multi-tenancy and mass-scaling.

<https://wiki.openstack.org/wiki/Neutron>

<https://docs.openstack.org/mitaka/install-guide-ubuntu/neutron-concepts.html>

OpenStack storage is ephemeral or persistent. When the administrator creates a VM with Nova, he does not have access to permanent storage. Only in ephemeral. This means that if the VM shuts down, the data are lost. The permanent storage is independent of VM operation and always available.

A form of permanent storage is that of objects. It is implemented on the platform by Swift (Object Storage service). It is used to store or archive large data sets.

It uses RESTfull API. Users have access through Horizon UI while Keystone authentication is used.

It is a collection of binary files with a unique identity. It is ideal for storing of the images of VMs while it supports asynchronous copying of data between different data centers. It offers easy scaling, high availability and data integrity because they are done with code rather than hardware. The files are recorded on disks on various servers and the program takes over for the rest, regardless of the hardware technology. If a hardware failure occurs, Swift restores the data from other drives.

<https://docs.openstack.org/arch-design/design-storage/design-storage-concepts.html>

Another form of permanent storage is Block storage. It is implemented by Cinder (Block Storage service). It is supported by drivers that enable the VM to access the drives directly, ensuring faster speed and better response. It is the ideal format for applications that these things matter, such as databases, etc. NFS, GlusterFS, and other file systems are also supported. They can be integrated into a VM in the form of a virtual disk. Users manage the drives through Compute and Dashboard. The creation of snapshots is supported for security while storage systems of many manufacturers can be used.

<https://docs.openstack.org/arch-design/design-storage/design-storage-concepts.html>

In multi-user environments, Shared File Systems service called Manila can be used. It provides shared file systems management. These are exported from share servers, that is, special VMs and are based on protocols such as NFS, CIFS, GlusterFS, or HDFS. Manila is a permanent storage space and can be integrated into any VM the administrator wants. The administrator defines the size, who has access to it or knows its existence, rules and limits of size and speed of access while he can make copies and snapshots for security.

<https://docs.openstack.org/arch-design/design-storage/design-storage-concepts.html>

Glance offers the images for the VMs. Images are virtual copies of hard disks and can contain bootable discs with an operating system or just discs with files. These images are used to create new VMs. Glance can also create machine replicas. It provides a RESTfull API with which the user can discover and manage the images.

<https://docs.openstack.org/glance/pike/>

Keystone (Identity) offers authentication and authorization services for all OpenStack services. It also is, of course, a directory of these services. It is the central directory of all users where their access rights to the services are registered. It can integrate directory services such as LDAP. Authentication services can have many forms, such as a combination of username and password, token-based systems and AWS compliant.

Horizon (Dashboard) is the graphical management interface of the platform. With Horizon, users have access to all OpenStack components, including third party add-ons. They can also monitor the state of the infrastructure. The OpenStack API or an EC2-compliant API can be used.

<https://opensource.com/resources/what-is-openstack>

<https://en.wikipedia.org/wiki/OpenStack>

Watcher is a flexible and scalable resource optimization service for an OpenStack cloud. It includes a metrics receiver, an optimizing processor, and an action plan applicator. With Watcher, we can achieve a reduction in the cost of running the data center and increasing performance by transferring VMs to another physical server.

In addition to embedded optimization methods, we can add new customized algorithms, metrics, and data processors. There are 2 modes of operation, the advice mode, which is manual and the active mode, which is automatic. With the first one the administrator is informed of what can be done and if he wants he can let it run while with the second he lets the system act automatically.

With use, the load of VMs varies. Watcher can transfer VMs from a high-load server to another with a lower one to reduce consumption or vice versa if someone is has a small number of VMs and its resources are not fully exploited. It may also terminate the operation of a server with few VMs by transferring them to another or launching one if the active ones are not sufficient for the load. This way we have significant energy savings.

<https://wiki.openstack.org/wiki/Watcher>

<http://specs.openstack.org/openstack/watcher-specs/specs/pike/implemented/energy-saving-strategy.html>

Another tool for this purpose is the OpenStack Neat of the Melbourne University's Cloud Computing and Distributed Systems (CLOUDS) lab (Anton Beloglazov and Rajkumar Buyya (2014)):

<http://openstack-neat.org/>

There is a general perception that open source software and open code technologies are lagging behind in security and vulnerabilities are not tackled effectively. Many companies that are using OpenStack to provide public clouds, as well as large customers such as BMW, Disney, Walmart, eBay, Paypal, the National Security Agency (NSA) etc. that they have implemented their own private clouds with OpenStack, help developing platform's security.

In fact, the opposite is the case. As we have seen in other open-source software such as Linux and Apache, bugs are easier and faster to deal with because many more people deal with them. Thousands of developers are involved in both the basic system and the extra tools of the platform development. All of them are backed up and guided by four security-aware teams, the OpenStack Security Project, security experts for each tool, commercial clouds manufacturers-providers and the basic OpenStack developers.

The OpenStack Security Project incorporates the Vulnerability Management Team (VMT) that discovers and solves security issues and vulnerabilities of platform segments. VMT provides code and architecture improvements and related documentation for developers. It issues the OpenStack Security Guide, the Advisories and the Notes. The OpenStack Security Guide includes best security practices for all segments for those who want to implement an OpenStack cloud.

<https://docs.openstack.org/security-guide/>

Whenever a vulnerability is discovered and resolved, the VMT issues a Security Advisory. These are being accumulated in Notes, along with tips for third-party tools and configuration issues. The experts for each tool deal exclusively with it and work with the VMT to solve the problems. The same is true for commercial clouds providers as it is in the best interests of them the product to be as safe as possible.

The OpenStack Security Project publishes guidelines and best practices for developers. Besides, however, it has developed tools and automations that help in security. Bandit is a standalone tool that detects errors in Python (Rajyalakshmi Marathu, Divya K Konoor and Prashanth Reddy (2016)). Syntribos detects security issues in RESTfull APIs and services while Anchor is an ephemeral PKI certification authority and provides short-term certificates.

The most critical elements of a cloud that need security are administration, communications, APIs, identification, control panel, VM management, storage, network, control messages, data processing, databases, privacy of customer data, VM security management, operation monitoring and compliance with legislation.

OpenStack supports multi-factor authentication, beyond the username and password combination. The keystone supports LDAP and external methods such as unified identification with identity providers (David W. Chadwick et al., (2013)).

Encryption of data in storage is supported. Encrypted data remain encrypted even when it is transmitted over a network. The capabilities of many protocols such as iSCSI are used.

So far, the OpenStack platform has complied with the most important security and personal data management rules.

<https://www.openstack.org/assets/securing-openstack-clouds/OpenStack-SecurityBrief-letteronline.pdf>

<https://docs.openstack.org/security-guide/compliance.html>

Table 1: Evaluation of the 4 IaaS solutions that are presented concerning the 5 characteristics/criteria that are explored

	Elasticity	Security	Cost efficiency / Energy savings	Availability	Vendor lock in
Amazon	Horizontal*	High	High	High	Average
Google Compute Engine	Horizontal*	High	High	High	Average to Low
Microsoft Azure	Horizontal*	High	Average	High	Low
OpenStack	Horizontal*	High	High	Depends on implementation	Not exist

* Vertical elasticity is provided by all solutions but requires shutting down, changing type, and restarting the virtual machine.

5 OpenStack presentation

For a presentation of the capabilities of the OpenStack platform we will use the DevStack simulation.

In order not to have trouble with the local network of CERTH we created a Virtual Machine in VMware Workstation 12 and used NAT for the network. This way we can also have versioning of the installation and snapshots that give us the capability to return the installation to previous states if wanted.

First we make a clean, minimal install of Ubuntu 16.04 Server LTS in a new VM. We use this edition because it is the most tested with DevStack.

After the cloning of the DevStack package we make a text file named `local.conf` in the `devstack` directory. The contents of the file are:

```
[[local|localrc]]  
  
FLOATING_RANGE=192.168.229.224/27  
  
FIXED_RANGE=10.0.0.0/26  
  
FIXED_NETWORK_SIZE=256  
  
FLAT_INTERFACE=ens33  
  
ADMIN_PASSWORD=secret  
  
DATABASE_PASSWORD=secret  
  
RABBIT_PASSWORD=secret  
  
SERVICE_PASSWORD=secret  
  
VIRT_DRIVER=libvirt  
  
LIBVIRT_TYPE=qemu
```

The `FLOATING_RANGE` is the range of the floating IPs we will use for the communication of the VMs with other networks. It is the network named “public”. The `FIXED_RANGE` is the range of the IPs of the internal network that the VMs will use for communication in it. It can be divided to subnets. It is the network named “private”.

A client can have one or more subnets. The FLAT_INTERFACE is the interface of the Ubuntu's network card. Then we define the same password for all the DevStack services for convenience. Finally we define the type of virtualization for the VMs as qemu because only this type has no problems with VMware Workstation. It is the only type that provides working nested virtualization.

Then we start the install of DevStack running the stack.sh script.

```
apt-get-update      6
pip_install        430
osc                121
wait_for_service   28
git_timed          437
dbsync             34
apt-get            409
-----
Unaccounted time   399
=====
Total runtime      1891

This is your host IP address: 192.168.229.129
This is your host IPv6 address: ::1
Horizon is now available at http://192.168.229.129/dashboard
Keystone is serving at http://192.168.229.129/identity/
The default users are: admin and demo
The password: secret

WARNING:
Using lib/neutron-legacy is deprecated, and it will be removed in the future

Services are running under systemd unit files.
For more information see:
https://docs.openstack.org/devstack/latest/systemd.html

DevStack Version: rocky
Change: 56225e19fe3e7064d635f4fe5684f6e1d36192f7 Merge "fix typo in python3_version" 2018-06-22 15:28:37 +0000
OS Version: Ubuntu 16.04 xenial

2018-06-23 11:18:51.405 | stack.sh completed in 1891 seconds.
stack@OpenStack02:~/devstack$
stack@OpenStack02:~/devstack$ _
```

Image 8: The installation of DevStack is complete

If we want to use the OpenStack CLI we install it first and then we download through Horizon the appropriate rc file of the project we want to work with and source it to make the configuration available.

```
source admin-openrc.sh
```

This is Horizon's first screen:

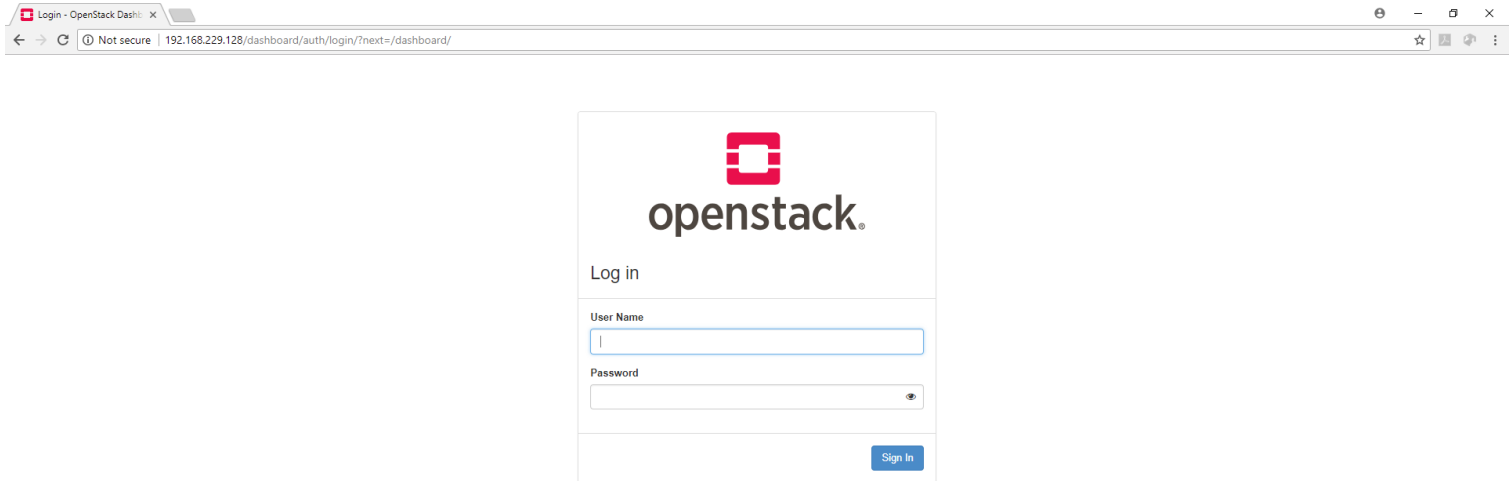


Image 9: Horizon log in screen

The Log in screen:

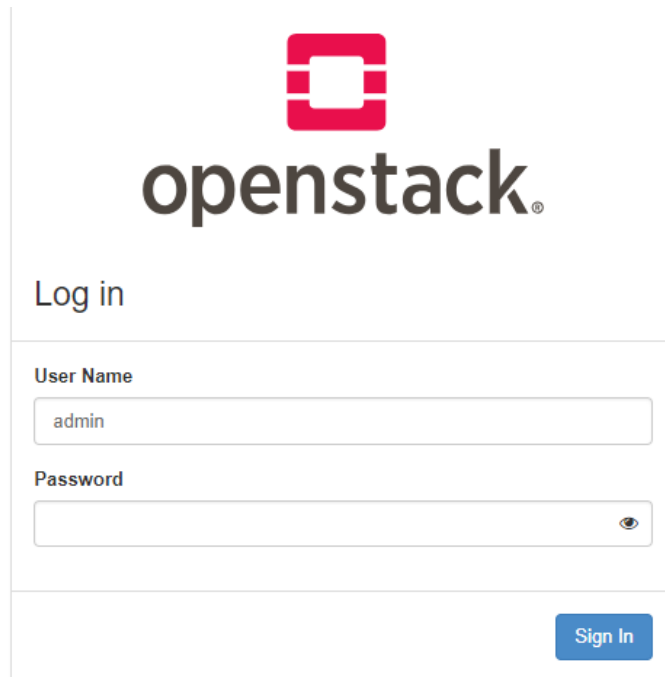


Image 10: Sign in as admin

From the upper left drop down menu we select the project “demo”. We can see all the projects that are created by the DevStack setup.

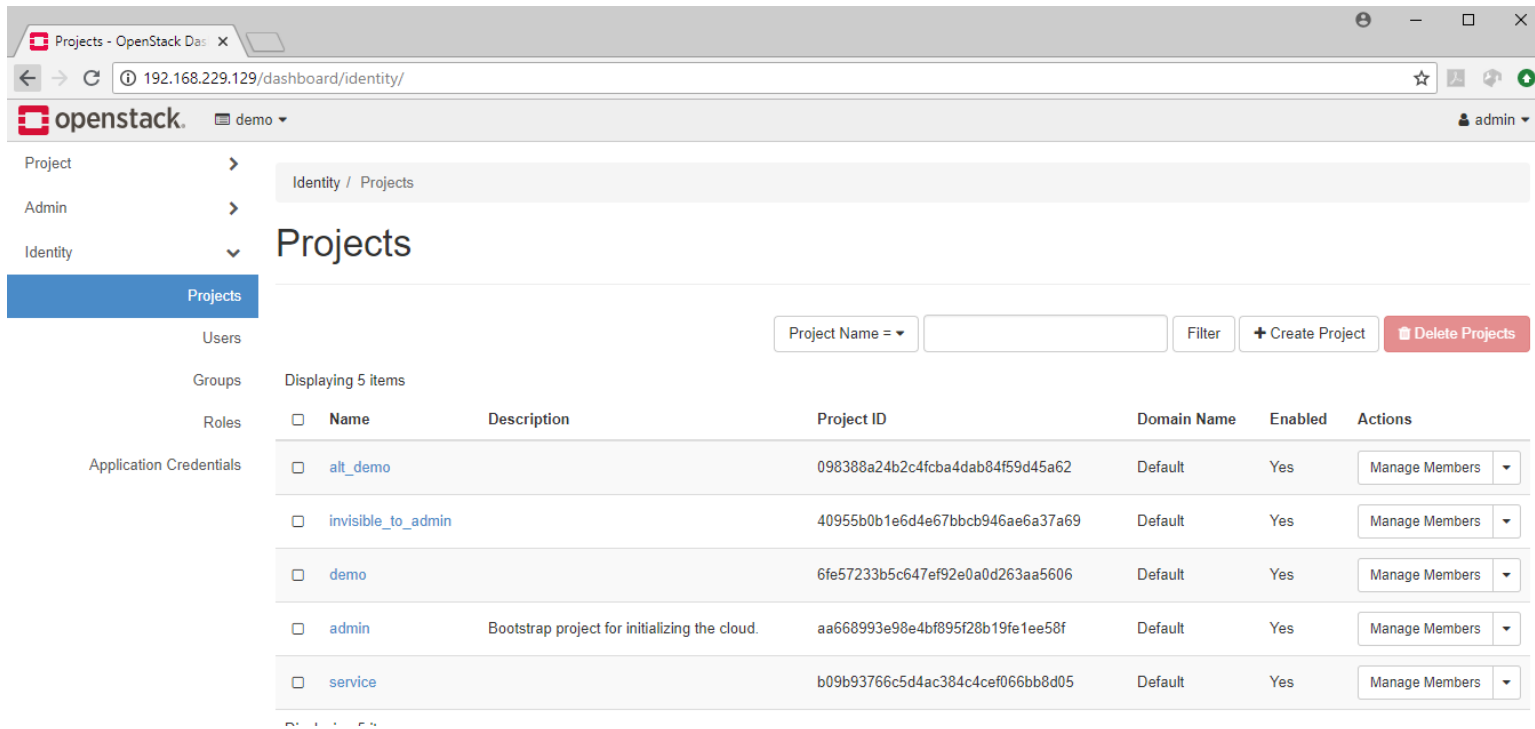


Image 11: The initial projects installed by Devstack

Next we can see the initial network topology of the project “demo” with the “public” and the “private” networks and the router that connects them. The “public” network will be available for all the projects of the cloud.

← → ↻ ⓘ 192.168.229.129/dashboard/project/network_topology/

openstack. demo ▾

Project ▾

API Access

Compute >

Volumes >

Network ▾

Network Topology

Networks

Routers

Security Groups

Floating IPs

Admin >

Identity >

Project / Network / Network Topology

Network Topology

Topology Graph

Small Normal

The diagram illustrates a network topology with two vertical bars representing networks. The left bar is blue and labeled 'public' with the IP address '2001:db8::64'. The right bar is orange and labeled 'private' with the IP address 'fd25:fe3d:7caf::64'. A small square icon with a cross inside, representing a router, is positioned between the two bars, connected to both by thin lines.

Launch Instance + Create Network + Create Router

Image 12: The initial network topology created during installation for the project “demo”

<https://docs.openstack.org/horizon/latest/user/configure-access-and-security-for-instances.html>

5.1 Security groups

The Security groups are the firewall that any client of the cloud can have. The VMs have access only to each other within a project (the project of the client) unless the administrator of the project defines other rules.

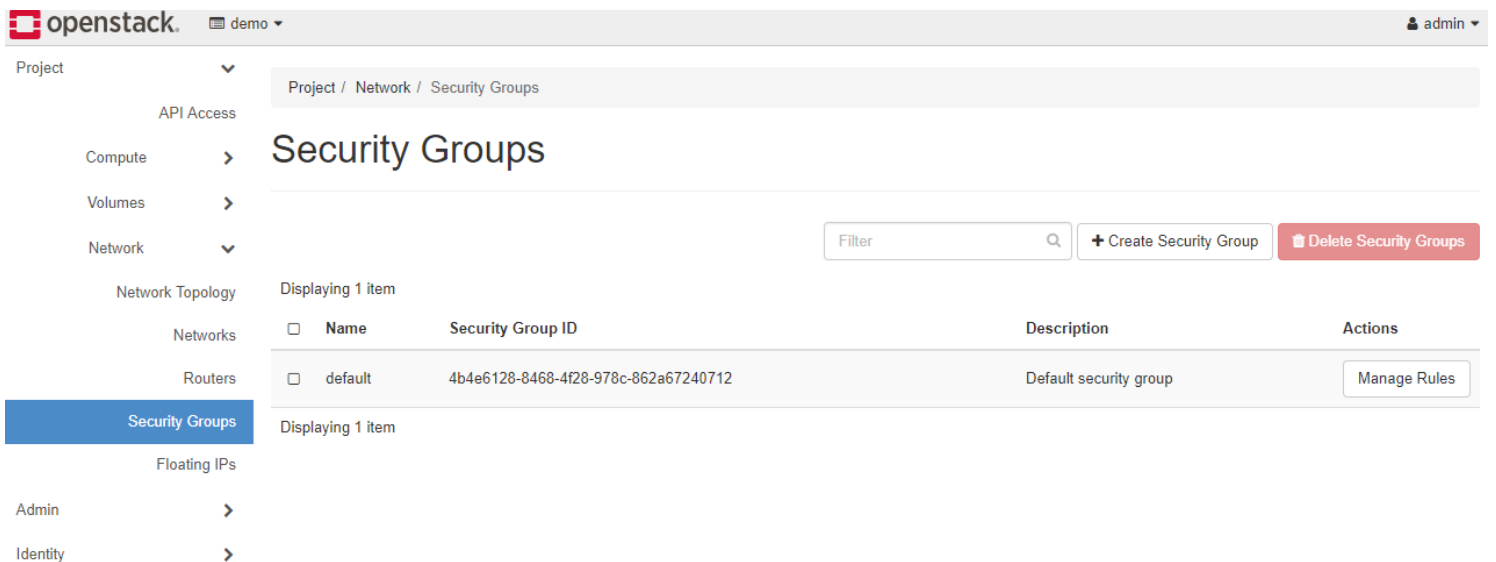


Image 13: The Security Groups with the preinstalled default one

- Project
- API Access
- Compute
- Volumes
- Network
 - Network Topology
 - Networks
 - Routers
 - Security Groups**
 - Floating IPs
- Admin
- Identity

Project / Network / Security Groups / Manage Security Group Rules

Manage Security Group Rules: default (6a8456ef-6af4-44b3-bf10-763f0fc7c1f5)

[+ Add Rule](#) [Delete Rules](#)

Displaying 4 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	:::0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	Any	Any	-	default	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv6	Any	Any	-	default	-	Delete Rule

Displaying 4 items

Image 14: The rules of the “default” security group

The next step is to create new security groups. We will create one security group for the Linux VMs that will enable SSH connection and ICMP communication inwards and outwards and another for the MS Windows VMs that will enable Remote Desktop (RDP) connection and ICMP communication inwards and outwards too.

5.1.1 New Security Group for SSH and ICMP (ping) in Linux VMs

5.1.1.1 Security Group creation

Create Security Group ✕

Name *
SSH and ICMP (ping) for Linux VMs

Description
SSH and ICMP (ping) for Linux VMs

Description:
Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group.

Cancel Create Security Group

Image 15: The creation of a new security group for the Linux VMs. It allows SSH connections and ICMP (ping) communication

This procedure enables SSH and ICMP (ping) access to instances. The rules may apply to all instances within a given project, and should be set for every project unless there is a reason to prohibit SSH or ICMP access to the instances.

5.1.1.2 Rule addition to a security group:

This procedure can be adjusted as necessary to add additional security group rules to a project, if the client requires them.

Here we manage the security group rules:

Project / Network / Security Groups / Manage Security Group Rul...

Manage Security Group Rules: SSH and ICMP (ping) for Linux VMs (b8874846-8a20-4b65-a522-50f7f0a27089)

Displaying 2 items

[+ Add Rule](#) [Delete Rules](#)

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	-	Delete Rule

Displaying 2 items

Image 16: The initial rules of the new security group

We select the button “Add Rule” to add SSH:

Add Rule ✕

Rule *
SSH

Description ⓘ
SSH

Remote * ⓘ
Security Group

Security Group
SSH and ICMP (ping) for Linux VMs (current)

Ether Type
IPv4

Description:
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the “Port Range” option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

[Cancel](#) [Add](#)

Image 17: Addition of the rule for SSH connections

With the same procedure we add the ICMP (ping) rule.

First Ingress:

Add Rule ✕

Rule *

All ICMP ▼

Description ?

ICMP (ping) Ingress

Direction

Ingress ▼

Remote * ?

Security Group ▼

Security Group

SSH and ICMP (ping) for Linux VMs (current) ▼

Ether Type

IPv4 ▼

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel Add

Image 18: Addition of the rule for ICMP inwards

Then Egress:

Add Rule



Rule *

All ICMP

Description ⓘ

ICMP (ping) Egress

Direction

Egress

Remote * ⓘ

Security Group

Security Group

SSH and ICMP (ping) for Linux VMs (current)

Ether Type

IPv4

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel

Add

Image 19: Addition of the rule for ICMP outwards

Manage Security Group Rules: SSH and ICMP (ping) for Linux VMs (b8874846-8a20-4b65-a522-50f7f0a27089)

+ Add Rule

Delete Rules

Displaying 5 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	ICMP	Any	-	SSH and ICMP (ping) for Linux VMs	ICMP (ping) Egress	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	-	SSH and ICMP (ping) for Linux VMs	ICMP (ping) Ingress	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	22 (SSH)	-	SSH and ICMP (ping) for Linux VMs	SSH	Delete Rule

Displaying 5 items

Image 20: The final rules of the security group for the Linux VMs

5.1.2 New Security Group for RDP and ICMP (ping) in MS Windows VMs

5.1.2.1 Security Group creation

Create Security Group ✕

Name *

RDP and ICMP (ping) for MS Windows VMs

Description

RDP and ICMP (ping) for MS Windows VMs

Description:

Security groups are sets of IP filter rules that are applied to network interfaces of a VM. After the security group is created, you can add rules to the security group.

Cancel

Create Security Group

Image 21: The creation of a new security group for the MS Windows VMs. It allows Remote Desktop connections and ICMP communication

Security Groups

Filter



+ Create Security Group

Delete Security Groups

Displaying 3 items

<input type="checkbox"/>	Name	Security Group ID	Description	Actions
<input type="checkbox"/>	RDP and ICMP (ping) for MS Windows VMs	523f5c94-8232-40ed-8b37-a9574587cd7f	RDP and ICMP (ping) for MS Windows VMs	Manage Rules ▾
<input type="checkbox"/>	SSH and ICMP (ping) for Linux VMs	b8874846-8a20-4b65-a522-50f7f0a27089	SSH and ICMP (ping) for Linux VMs	Manage Rules ▾
<input type="checkbox"/>	default	6a8456ef-6af4-44b3-bf10-763f0fc7c1f5	Default security group	Manage Rules

Displaying 3 items

Image 22: The 3 security groups

5.1.2.2 Rule addition to a security group:

We add RDP with the same procedure:

Add Rule ✕

Rule *

RDP ▾

Description ⓘ

RDP for MS Windows VMs

Remote * ⓘ

Security Group ▾

Security Group

RDP and ICMP (ping) for MS Windows VMs (cur... ▾)

Ether Type

IPv4 ▾

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel Add

Image 23: Addition of the rule for RDP connections

We add ICMP (ping) Ingress:

Add Rule ✕

Rule *

All ICMP ▼

Description ⓘ

ICMP (ping) for Windows VMs Ingress

Direction

Ingress ▼

Remote * ⓘ

Security Group ▼

Security Group

RDP and ICMP (ping) for MS Windows VMs (cur... ▼

Ether Type

IPv4 ▼

Description:

Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:

Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.

Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.

Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Image 24: Addition of the rule for ICMP inwards

And ICMP (ping) Egress:

Add Rule ✕

Rule *
All ICMP ▼

Description ?
ICMP (ping) for MS Windows VMs Egress

Direction
Egress ▼

Remote * ?
Security Group ▼

Security Group
RDP and ICMP (ping) for MS Windows VMs (cur... ▼

Ether Type
IPv4 ▼

Description:
Rules define which traffic is allowed to instances assigned to the security group. A security group rule consists of three main parts:
Rule: You can specify the desired rule template or use custom rules, the options are Custom TCP Rule, Custom UDP Rule, or Custom ICMP Rule.
Open Port/Port Range: For TCP and UDP rules you may choose to open either a single port or a range of ports. Selecting the "Port Range" option will provide you with space to provide both the starting and ending ports for the range. For ICMP rules you instead specify an ICMP type and code in the spaces provided.
Remote: You must specify the source of the traffic to be allowed via this rule. You may do so either in the form of an IP address block (CIDR) or via a source group (Security Group). Selecting a security group as the source will allow any other instance in that security group access to any other instance via this rule.

Cancel Add

Image 25: Addition of the rule for ICMP outwards

Manage Security Group Rules: RDP and ICMP (ping) for MS Windows VMs (523f5c94-8232-40ed-8b37-a9574587cd7f)

+ Add Rule

Delete Rules

Displaying 5 items

<input type="checkbox"/>	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
<input type="checkbox"/>	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
<input type="checkbox"/>	Egress	IPv4	ICMP	Any	-	RDP and ICMP (ping) for MS Windows VMs	ICMP (ping) for MS Windows VMs Egress	Delete Rule
<input type="checkbox"/>	Egress	IPv6	Any	Any	::/0	-	-	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	ICMP	Any	-	RDP and ICMP (ping) for MS Windows VMs	ICMP (ping) for Windows VMs Ingress	Delete Rule
<input type="checkbox"/>	Ingress	IPv4	TCP	3389 (RDP)	-	RDP and ICMP (ping) for MS Windows VMs	RDP for MS Windows VMs	Delete Rule

Displaying 5 items

Image 26: The final rules of the security group for the MS Windows VMs

5.2 Key Pairs

A key pair is needed for SSH connections.

5.2.1 Creation:

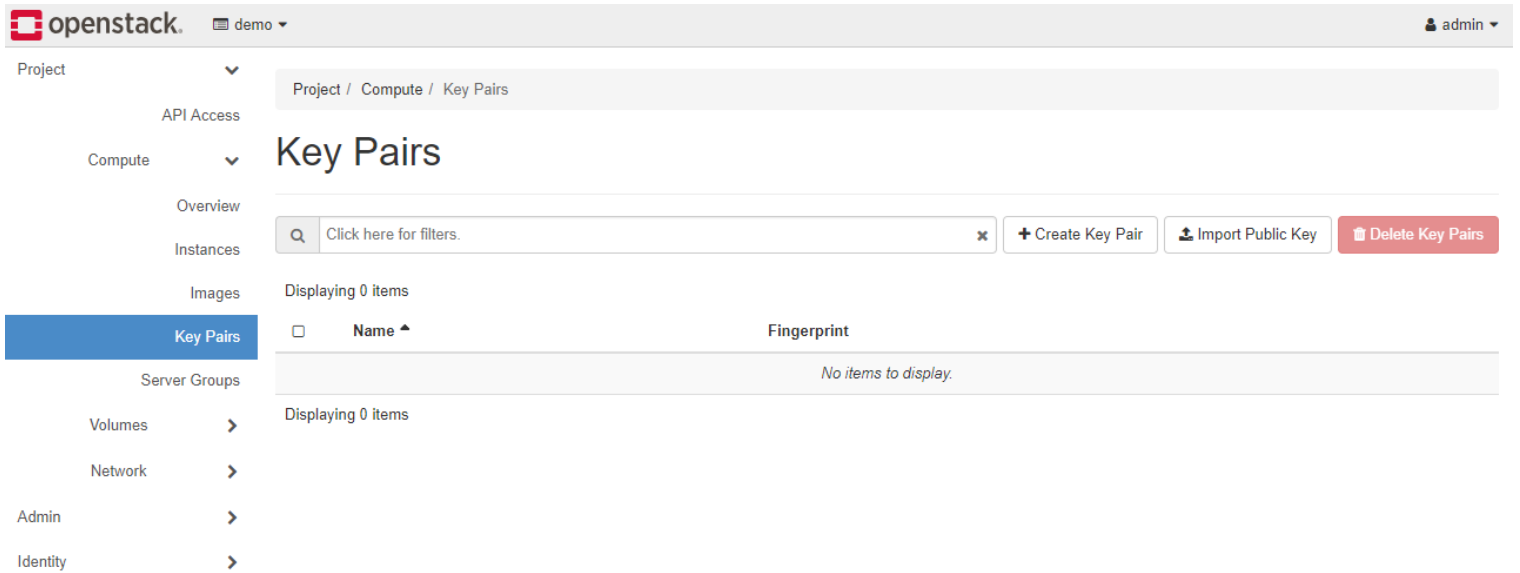


Image 27: The screen for key pair management with no key pairs created

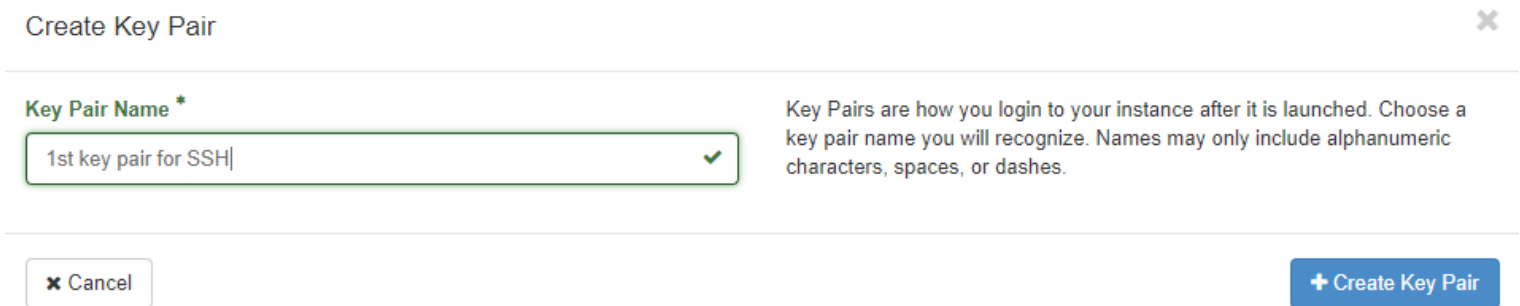


Image 28: The creation of a key pair for SSH connections

Key Pairs

✕
+ Create Key Pair
📄 Import Public Key
🗑️ Delete Key Pairs

Displaying 1 item

<input type="checkbox"/>	Name ^	Fingerprint	
<input type="checkbox"/>	> 1st key pair for SSH	9d:15:c6:d3:35:d1:db:cc:c0:d5:0a:c9:f1:87:86:5b	🗑️ Delete Key Pair

Displaying 1 item

Image 29: The first key pair shown in the key pair management screen

5.3 Images

The images we upload as admin are available to all projects.

Here we see all the images:

Project / Compute / Images

Images

✕
+ Create Image
🗑️ Delete Images

Displaying 1 item

<input type="checkbox"/>	Owner	Name ^	Type	Status	Visibility	Protected	Disk Format	Size	
<input type="checkbox"/>	> admin	cirros-0.3.5-x86_64-disk	Image	Active	Public	No	QCOW2	12.65 MB	Launch ▼

Displaying 1 item

Image 30: The images management screen with the preinstalled cirros image

Next we create a new image. It will be Ubuntu 16.04 server 64 bit LTS. We upload it to Glance.

Create Image ✕

Image Details *

Metadata

?

Image Details

Specify an image to upload to the Image Service.

Image Name *

Image Description

Image Source

File *

Format *

Image Requirements

Kernel

Architecture

Ramdisk

Minimum Disk (GB)

Minimum RAM (MB)

Image Sharing

Visibility

Protected

Image 31: First screen of the upload of a new image

Image Details

Specify an image to upload to the Image Service.

Image Name*

Image Description

Image Source

File*

Browse... ubuntu-16.04-server-cloudimg-amd64-

Format*

QCOW2 - QEMU Emulator

Image Requirements

Kernel

Choose an image

Ramdisk

Choose an image

Architecture

Minimum Disk (GB)

Minimum RAM (MB)

Image Sharing

Visibility

Public Private

Protected

Yes No

✕ Cancel

< Back

Next >

✓ Create Image

Image 32: Second screen of the upload of an image. It will be Ubuntu 16.04 server 64 bit LTS. We select QCOW2-QEMU emulation, AMD64 architecture and we define it as Public to be generally shared and with no protection

Next we specify resource metadata:

Create Image ✕

Image Details ?

Metadata

You can specify resource metadata by moving items from the left column to the right column. In the left column there are metadata definitions from the Glance Metadata Catalog. Use the "Custom" option to add metadata with the key of your choice.

Available Metadata

- Custom
- › CIM Processor Allocation Setting
- › Cinder Volume Type
- › Common Operating System Properties
- › CPU Pinning
- › Database Software
- › Guest Memory Backing
- › Hypervisor Selection
- › Image Signature Verification

Existing Metadata

No existing metadata

Click each item to get its description here.

✕ Cancel< BackNext >✓ Create Image

Image 33: First screen of the available metadata for the image. There are many options available

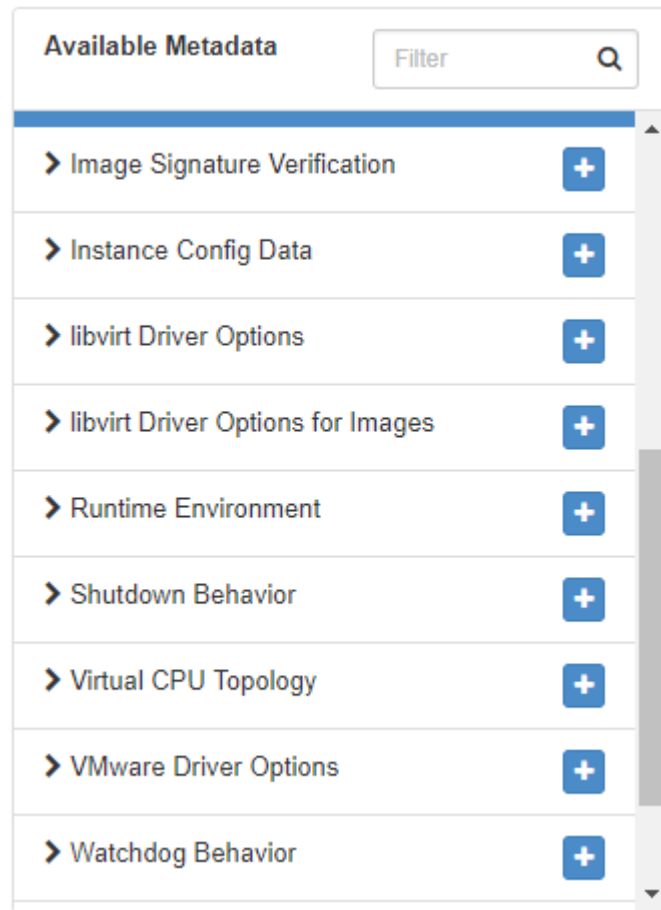


Image 34: Second screen of the available metadata for the image

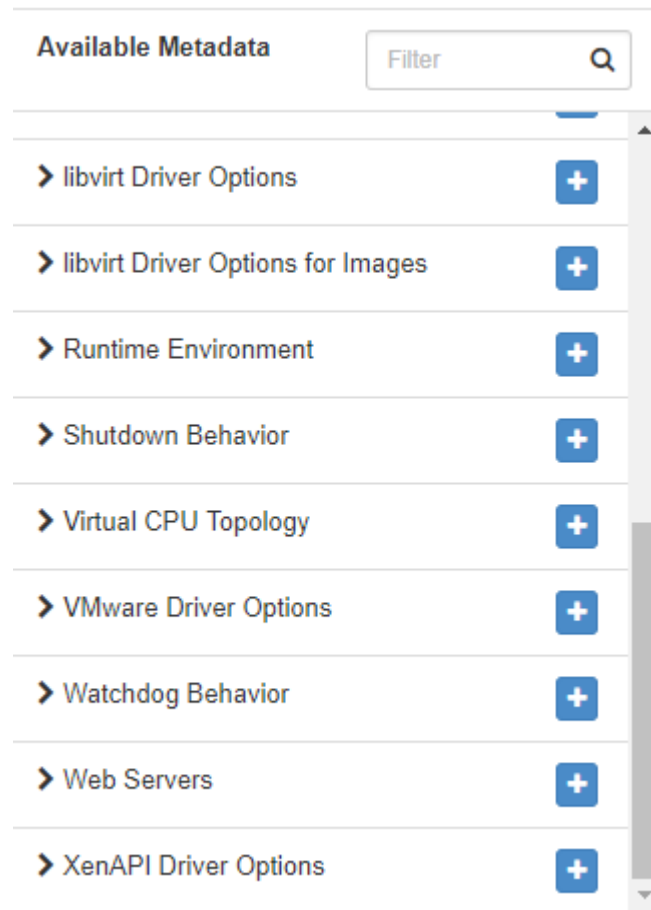


Image 35: Third screen of the available metadata for the image

The next step is to upload a Windows Server 2012 R2 Std Eval image to Glance:

Create Image ✕

Image Details ?

Specify an image to upload to the Image Service.

Image Name*

Image Description

Image Source

File*

Format*

Image Requirements

Kernel

Ramdisk

Architecture

Minimum Disk (GB) **Minimum RAM (MB)**

Image Sharing

Visibility

Protected

Image 36: Another upload of an image. It will be MS Windows Server 2012 R2 Standard evaluation. We select QCOW2-QEMU emulation, x64 architecture and we define it as Public to be generally shared and with no protection

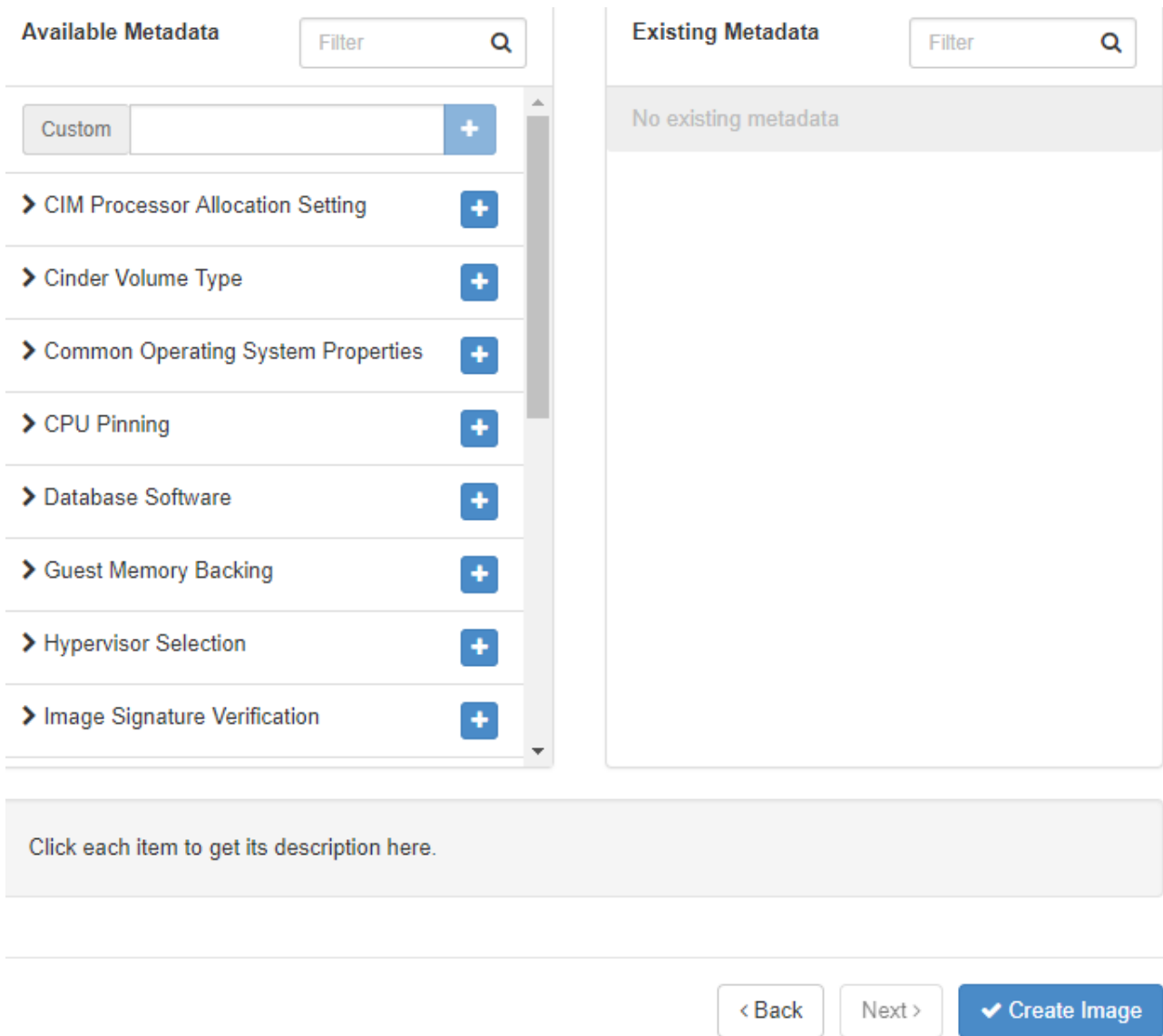
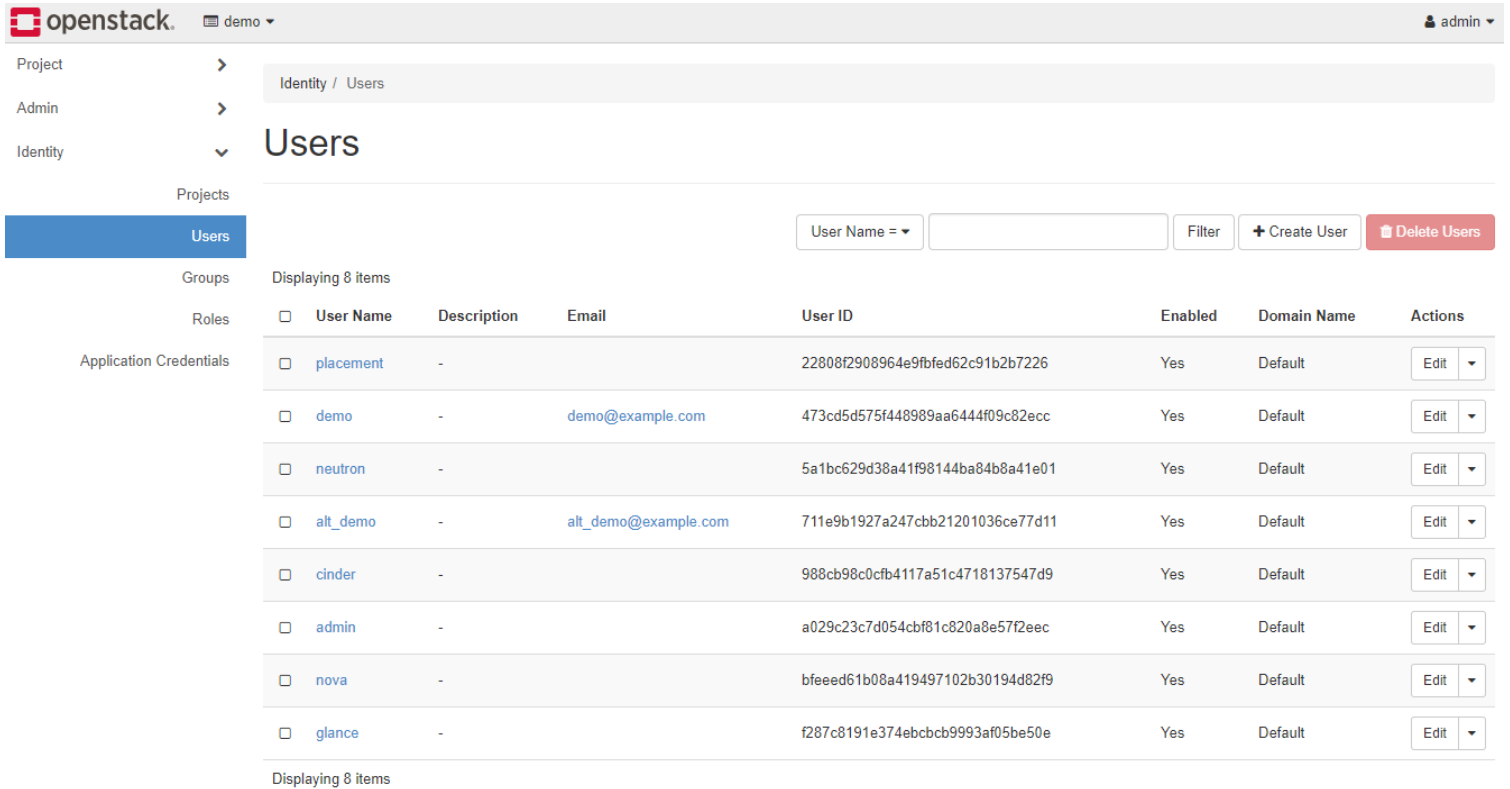


Image 37: We can again select the various metadata

5.4 User management

The admin user can create new users with different privileges. Each project must have its own user/administrator that will create and manage it.



The screenshot shows the OpenStack user management interface. The top navigation bar includes the OpenStack logo, a 'demo' dropdown, and a user profile for 'admin'. The left sidebar shows a navigation menu with 'Project', 'Admin', and 'Identity' sections. Under 'Identity', 'Users' is selected. The main content area displays a table of users with columns for 'User Name', 'Description', 'Email', 'User ID', 'Enabled', 'Domain Name', and 'Actions'. There are 8 users listed, including 'placement', 'demo', 'neutron', 'alt_demo', 'cinder', 'admin', 'nova', and 'glance'. Each user has an 'Edit' button. Above the table, there is a search filter for 'User Name', a 'Filter' button, and buttons for '+ Create User' and 'Delete Users'. The text 'Displaying 8 items' is shown above and below the table.

<input type="checkbox"/>	User Name	Description	Email	User ID	Enabled	Domain Name	Actions
<input type="checkbox"/>	placement	-		22808f2908964e9fbfed62c91b2b7226	Yes	Default	Edit
<input type="checkbox"/>	demo	-	demo@example.com	473cd5d575f448989aa6444f09c82ecc	Yes	Default	Edit
<input type="checkbox"/>	neutron	-		5a1bc629d38a41f98144ba84b8a41e01	Yes	Default	Edit
<input type="checkbox"/>	alt_demo	-	alt_demo@example.com	711e9b1927a247cbb21201036ce77d11	Yes	Default	Edit
<input type="checkbox"/>	cinder	-		988cb98c0cfb4117a51c4718137547d9	Yes	Default	Edit
<input type="checkbox"/>	admin	-		a029c23c7d054cbf81c820a8e57f2eec	Yes	Default	Edit
<input type="checkbox"/>	nova	-		bfeeed61b08a419497102b30194d82f9	Yes	Default	Edit
<input type="checkbox"/>	glance	-		f287c8191e374ebcbcb9993af05be50e	Yes	Default	Edit

Image 38: The user management screen with the predefined users

5.4.1 User creation

Here we add a user named cl1_user for a new client named client1:

Create User ✕

Domain ID

Domain Name

User Name *

Description

Email

Password *

Confirm Password *

Primary Project

Role

Enabled

Image 39: We create a new user named cl1_user for the client named client1.

We define a password and can select the project the new user will be applied to

Users

User Name = ▾

Filter

+ Create User

Delete Users

Displaying 9 items

<input type="checkbox"/>	User Name	Description	Email	User ID	Enabled	Domain Name	Actions
<input type="checkbox"/>	placement	-		22808f2908964e9fbfed62c91b2b7226	Yes	Default	Edit ▾
<input type="checkbox"/>	cl1_user	1st user for the client named client1	cl1@client1.com	40da007b117b47eaad974bcdda3f46e7	Yes	Default	Edit ▾
<input type="checkbox"/>	demo	-	demo@example.com	473cd5d575f448989aa6444f09c82ecc	Yes	Default	Edit ▾
<input type="checkbox"/>	neutron	-		5a1bc629d38a41f98144ba84b8a41e01	Yes	Default	Edit ▾
<input type="checkbox"/>	alt_demo	-	alt_demo@example.com	711e9b1927a247cbb21201036ce77d11	Yes	Default	Edit ▾
<input type="checkbox"/>	cinder	-		988cb98c0cfb4117a51c4718137547d9	Yes	Default	Edit ▾
<input type="checkbox"/>	admin	-		a029c23c7d054cbf81c820a8e57f2eec	Yes	Default	Edit ▾
<input type="checkbox"/>	nova	-		bfeeed61b08a419497102b30194d82f9	Yes	Default	Edit ▾
<input type="checkbox"/>	glance	-		f287c8191e374ebc9993af05be50e	Yes	Default	Edit ▾

Displaying 9 items

Image 40: We see the new user in the user management screen

5.5 Project management

The admin can create a new project and the users that will manage it or only a new user/administrator that will create it. Here we see the first option.

The screenshot displays the OpenStack Identity web interface for managing projects. The top navigation bar shows the OpenStack logo, a 'demo' environment selector, and a user profile for 'admin'. The left sidebar contains a menu with 'Projects' selected. The main content area is titled 'Projects' and includes a search bar for 'Project Name', a 'Filter' button, and buttons for '+ Create Project' and 'Delete Projects'. Below this is a table listing 5 projects with columns for Name, Description, Project ID, Domain Name, Enabled, and Actions.

<input type="checkbox"/>	Name	Description	Project ID	Domain Name	Enabled	Actions
<input type="checkbox"/>	alt_demo		098388a24b2c4fcba4dab84f59d45a62	Default	Yes	Manage Members
<input type="checkbox"/>	invisible_to_admin		40955b0b1e6d4e67bbcb946ae6a37a69	Default	Yes	Manage Members
<input type="checkbox"/>	demo		6fe57233b5c647ef92e0a0d263aa5606	Default	Yes	Manage Members
<input type="checkbox"/>	admin	Bootstrap project for initializing the cloud.	aa668993e98e4bf895f28b19fe1ee58f	Default	Yes	Manage Members
<input type="checkbox"/>	service		b09b93766c5d4ac384c4cef066bb8d05	Default	Yes	Manage Members

Image 41: The project management screen with the predefined projects

5.5.1 Project creation

The admin adds a new project named client1 for a client of the cloud named client1:

Create Project ✕

Project Information * Project Members Project Groups

Domain ID: default

Domain Name: Default

Name *: client1

Description: 1st client (tenant) of the cloud

Enabled:

Cancel Create Project

Image 42: First screen of the creation of the new project named client1

Create Project



Project Information *

Project Members

Project Groups

All Users	Filter	Q
placement		+
demo		+
neutron		+
alt_demo		+
cinder		+
nova		+
glance		+

Project Members	Filter	Q
admin	Member	-
cl1_user	Member	-

Cancel Create Project

Image 43: We select the users that will be members of the new project

Create Project



Project Information *

Project Members

Project Groups

All Groups

nonadmins	<input data-bbox="683 488 761 571" type="button" value="+"/>
-----------	--

Project Groups

admins	<input data-bbox="1193 488 1348 571" type="button" value="Member"/> <input data-bbox="1348 488 1426 571" type="button" value="-"/>
--------	--

Cancel

Create Project

Image 44: We select the groups of the users that will be members of the new project

Identity / Projects

Projects

Project Name =

Filter

+ Create Project

Delete Projects

Displaying 6 items

<input type="checkbox"/>	Name	Description	Project ID	Domain Name	Enabled	Actions
<input type="checkbox"/>	alt_demo		098388a24b2c4fcb4dab84f59d45a62	Default	Yes	Manage Members
<input type="checkbox"/>	invisible_to_admin		40955b0b1e6d4e67bbcb946ae6a37a69	Default	Yes	Manage Members
<input type="checkbox"/>	demo		6fe57233b5c647ef92e0a0d263aa5606	Default	Yes	Manage Members
<input type="checkbox"/>	admin	Bootstrap project for initializing the cloud.	aa668993e98e4bf895f28b19fe1ee58f	Default	Yes	Manage Members
<input type="checkbox"/>	service		b09b93766c5d4ac384c4cef066bb8d05	Default	Yes	Manage Members
<input type="checkbox"/>	client1	1st client (tenant) of the cloud	ec80c304662244d9acbf1764a72de931	Default	Yes	Manage Members

Displaying 6 items

Image 45: The project management screen with the new project shown

5.6 Client1 project

Next we sign in with user cl1_user:


openstack[®]

Log in

User Name

Password

Image 46: We sign in as the new user cl1_user for the new project client1

Here we see an overview of the client1 project:

Overview

- Compute
- Overview**
- Instances
- Images
- Key Pairs
- Server Groups
- Volumes
- Network
- Identity

Limit Summary

Compute

- Instances: Used 0 of 10
- VCPU: Used 0 of 20
- RAM: Used 0Bytes of 50GB

Volume

- Volumes: Used 0 of 10
- Volume Snapshots: Used 0 of 10
- Volume Storage: Used 0Bytes of 1000GB

Network

- Floating IPs: Allocated 0 of 50
- Security Groups: Used 0 of 10
- Security Group Rules: Used 0 of 100
- Networks: Used 0 of 100
- Ports: Used 0 of 500
- Routers: Used 0 of 10

Usage Summary

Select a period of time to query its usage:
The date should be in YYYY-MM-DD format.

2018-06-30 to 2018-07-01 [Submit](#)

Active Instances: 0
Active RAM: 0Bytes
This Period's VCPU-Hours: 0.00
This Period's GB-Hours: 0.00
This Period's RAM-Hours: 0.00

Usage

Instance Name	VCPU	Disk	RAM	Time since created
No items to display.				

[Download CSV Summary](#)

Image 47: The overview of the project client1

5.6.1 The creation of a network for client1

Here we see the initial network topology of the new project. Only the network “public” is shown.

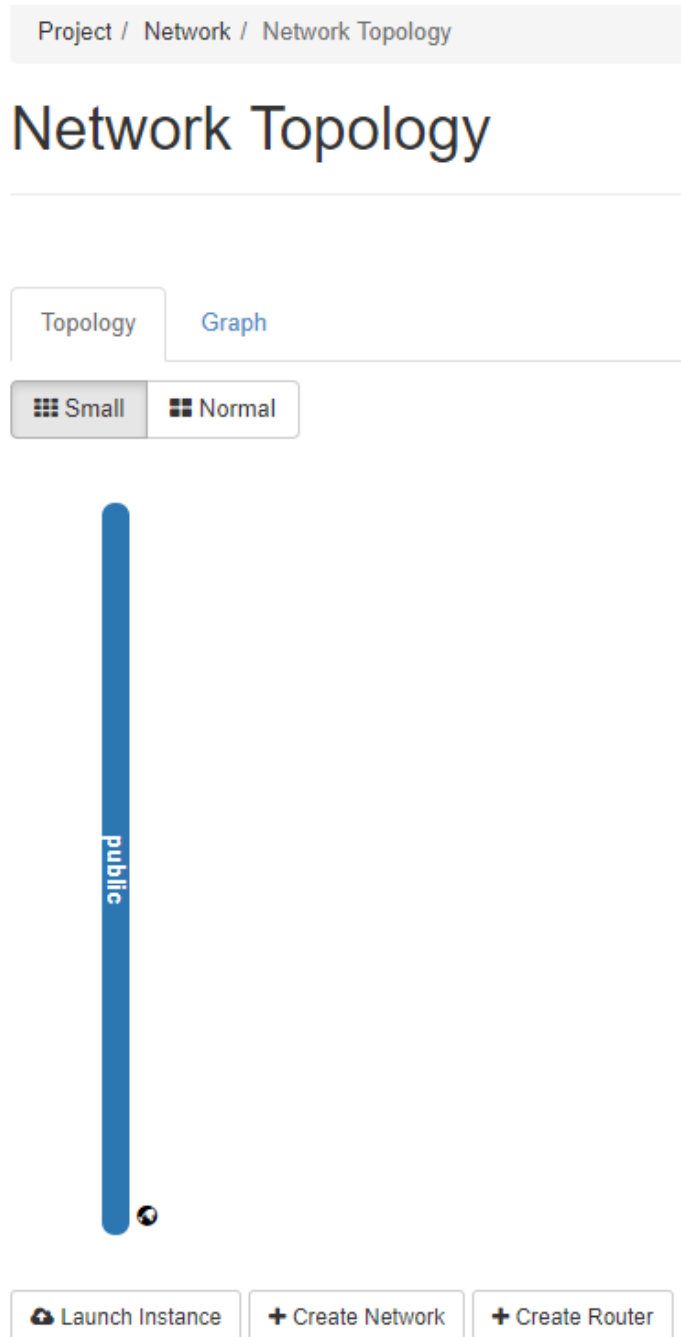


Image 48: The initial network topology of the new project client1. We see the same network named “public” that was created during installation

Next we create a new network named “private”, different from that of the project “demo”.

Create Network X

Network Subnet Subnet Details

Network Name

private|

Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Enable Admin State ?

Create Subnet

Availability Zone Hints ?

nova

Cancel « Back Next »

Image 49: The first screen of the creation of a new network named “private” for the client1 project

Create Network



Network

Subnet

Subnet Details

Subnet Name

private subnet for client1

Network Address Source

Enter Network Address manually

Network Address*

10.0.0.64/26

IP Version

IPv4

Gateway IP

10.0.0.65

Disable Gateway

Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab.

Cancel

« Back

Next »

Image 50: The definition of the subnet of the private network

Create Network



Network

Subnet

Subnet Details

Enable DHCP

Specify additional attributes for the subnet.

Allocation Pools ⓘ

DNS Name Servers ⓘ

Host Routes ⓘ

Cancel

« Back

Create

Image 51: The definition of the subnet details. We define the IP range, we enable DHCP for the range and the absence of a DNS server

Network Topology

Topology Graph

Small Normal

public private

10.0.0.64/26

Launch Instance Create Network Create Router

Image 52: The new network with the new subnet for the client1 project is shown in the network topology

Next we create a new router for the new project that will connect the new networks.

Create Router ✕

Router Name

Enable Admin State

External Network

Availability Zone Hints ⓘ

Description:
Creates a router with specified parameters.

Image 53: First screen of the creation of a new router named router2 for client1 project. It will connect the new “private” network of the project with the “public” one of the infrastructure

router2

Overview	Interfaces	Static Routes
Name	router2	
ID	25584604-baac-4f16-a637-9d2215e0b8aa	
Description		
Project ID	ec80c304662244d9acbf1764a72de931	
Status	Active	
Admin State	UP	
Availability Zones	<ul style="list-style-type: none"> nova 	

External Gateway

Network Name	public
Network ID	10da1227-9280-40de-8cce-44654866cee0
External Fixed IPs	<ul style="list-style-type: none"> Subnet ID 74d0266c-81c0-4fb9-b3ba-2c095eb2fd61 IP Address 192.168.229.232 Subnet ID bc325b40-88f2-4b7f-8a55-958ca93a3708 IP Address 2001:db8::b
SNAT	Enabled

Image 54: Overview of the new router

router2

Clear Gateway ▾

Overview Interfaces Static Routes

+ Add Interface

Name	Fixed IPs	Status	Type	Admin State	Actions
------	-----------	--------	------	-------------	---------

No items to display.

Image 55: We have not added any interfaces yet

Next we add an interface to connect the router to the new “private” network. The router is already connected to the available “public”.

Add Interface ✕

Subnet *
private: 10.0.0.64/26 (private subnet for client1) ▾

IP Address (optional) ⓘ
10.0.0.65

Description:
You can connect a specified subnet to the router.
If you don't specify an IP address here, the gateway's IP address of the selected subnet will be used as the IP address of the newly created interface of the router. If the gateway's IP address is in use, you must use a different address which belongs to the selected subnet.

Image 56: We add a new interface to the router router2. It will provide connection with the internal private network

Project / Network / Routers / router2

router2 Clear Gateway ▾

Overview Interfaces Static Routes

Displaying 1 item

<input type="checkbox"/>	Name	Fixed IPs	Status	Type	Admin State	Actions
<input type="checkbox"/>	(b214c555-b0aa)	• 10.0.0.65	Down	Internal Interface	UP	<input type="button" value="Delete Interface"/>

Displaying 1 item

Image 57: The new interface is shown

Network Topology

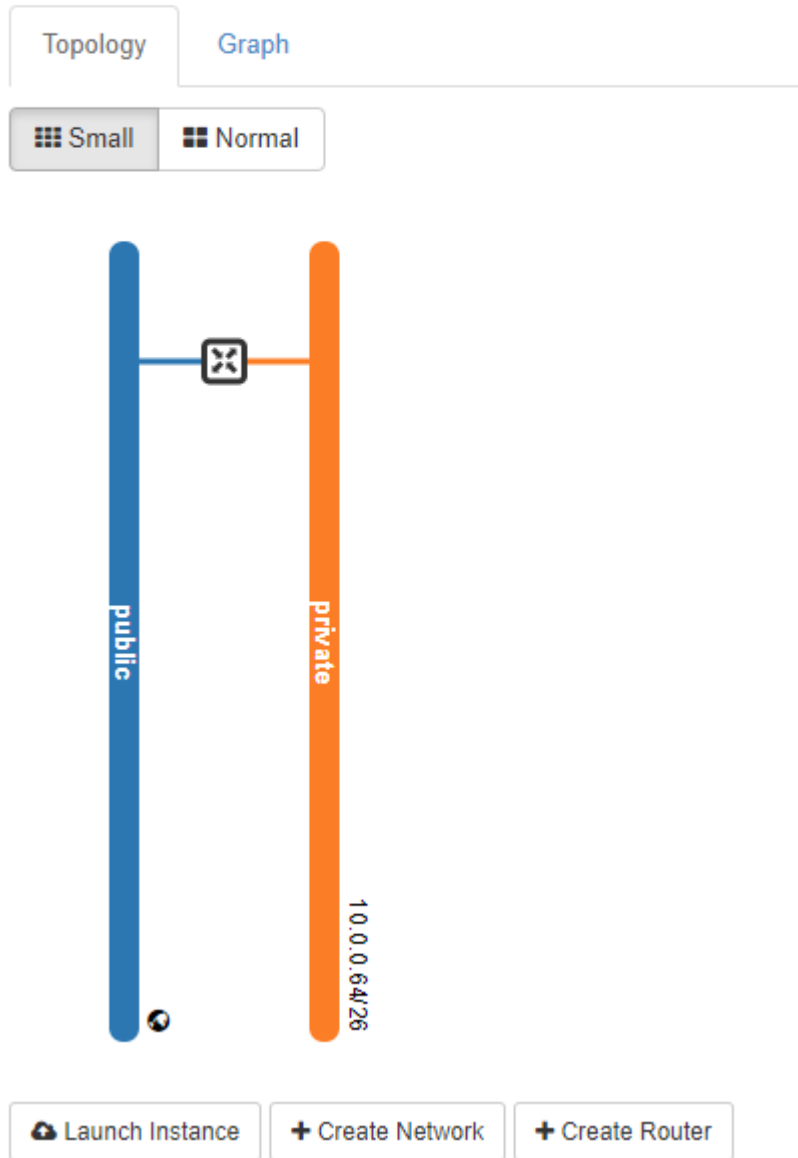


Image 58: The network topology with the 2 networks (public and private) connected via the router

5.6.2 The launch of an instance

Here we have an overview of the client1 project:

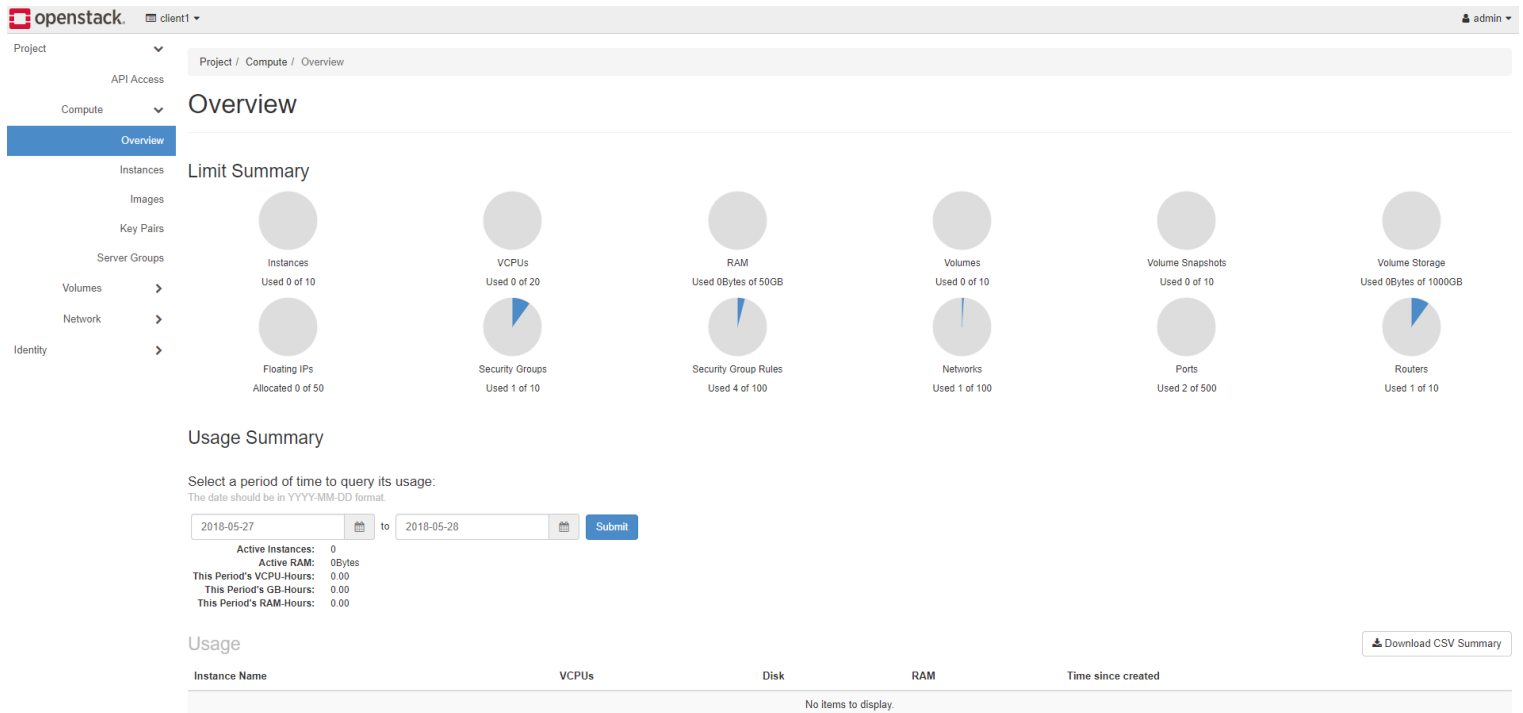


Image 59: Overview of the client1 project

From here we can launch an instance:

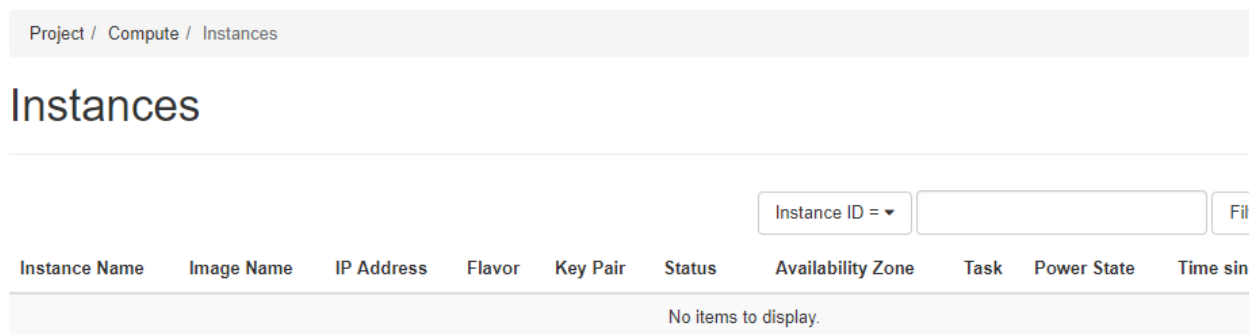


Image 60: The instances screen. We haven't created any yet

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name *

Description

Availability Zone

Count *

Total Instances (10 Max)

10%

- 0 Current Usage
- 1 Added
- 9 Remaining

Details

Source *

Flavor *

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

✕ Cancel

< Back

Next >

Launch Instance

Image 61: First screen of the creation of a new instance

Details

Source

Flavor *

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Instance source is the template used to create an instance. You can use an image, a snapshot of an instance (image snapshot), a volume or a volume snapshot (if enabled). You can also choose to use persistent storage by creating a new volume. ?

Select Boot Source

Image

Create New Volume

Yes No

Volume Size (GB) *

1

Delete Volume on Instance Delete

Yes No

Allocated

Name	Updated	Size	Type	Visibility
> cirros-0.3.5-x86_64-disk	6/23/18 9:09 PM	12.65 MB	qcow2	Public

Available 2

Select one

Click here for filters.

Name	Updated	Size	Type	Visibility
> MS Windows Server 2012 R2 Std Eval	6/30/18 2:14 PM	11.18 GB	qcow2	Public
> Ubuntu 1604 LTS	6/30/18 12:43 PM	278.50 MB	qcow2	Public

✕ Cancel

< Back

Next >

Launch Instance

Image 62: Second screen of the new instance creation. We use the light cirros image. We want a new volume for disk space to be created that will be deleted if we delete the instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Flavors manage the sizing for the compute, memory and storage capacity of the instance.



Allocated

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
> m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes	↓

Available 11

Select one

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public	
> m1.nano	1	64 MB	0 GB	0 GB	0 GB	Yes	↑
> m1.micro	1	128 MB	0 GB	0 GB	0 GB	Yes	↑
> cirros256	1	256 MB	0 GB	0 GB	0 GB	Yes	↑
> ds512M	1	512 MB	5 GB	5 GB	0 GB	Yes	↑
> ds1G	1	1 GB	10 GB	10 GB	0 GB	Yes	↑
> m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes	↑
> ds2G	2	2 GB	10 GB	10 GB	0 GB	Yes	↑
> m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes	↑
> ds4G	4	4 GB	20 GB	20 GB	0 GB	Yes	↑
> m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes	↑
> m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes	↑

✕ Cancel

< Back

Next >

Launch Instance

Image 63: We select the m1.tiny flavor among the many available

Launch Instance



Details

Networks provide the communication channels for instances in the cloud.



Source

▼ Allocated **1**

Select networks from those listed below.

Flavor

Networks

	Network	Subnets Associated	Shared	Admin State	Status	
1	private	private subnet for client1	No	Up	Active	↓

Network Ports

▼ Available **0**

Select at least one network

Security Groups

Key Pair

Network	Subnets Associated	Shared	Admin State	Status
No available items				

Configuration

Server Groups

Scheduler Hints

Metadata

✕ Cancel

< Back

Next >

Launch Instance

Image 64: The only available private network is preselected

Launch Instance



Details

Ports provide extra communication channels to your instances. You can select ports instead of networks or a mix of both.



Source

▼ Allocated

Select ports from those listed below.

Flavor

Networks

Name	IP	Admin State	Status
Select an item from Available items below			

Network Ports

▼ Available **0**

Select one

Security Groups

Key Pair

Name	IP	Admin State	Status
No available items			

Configuration

Server Groups

Scheduler Hints

Metadata

✕ Cancel

< Back

Next >

Launch Instance

Image 65: We haven't defined any network ports

Launch Instance



Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Select the security groups to launch the instance in.

▼ Allocated **2**

Name	Description	
> default	Default security group	↓
> SSH and ICMP (ping) for Linux VMs	SSH and ICMP (ping) for Linux VMs	↓

▼ Available **1** Select one or more

🔍 Click here for filters. ✕

Name	Description	
> RDP and ICMP (ping) for MS Windows VMs	RDP and ICMP (ping) for MS Windows VMs	↑

✕ Cancel < Back Next > Launch Instance

Image 66: We select the security group for Linux VMs that we want

Details

A key pair allows you to SSH into your newly created instance. You may select an existing key pair, import a key pair, or generate a new key pair.



Source

+ Create Key Pair

📁 Import Key Pair

Flavor

Allocated

Displaying 1 item

Name	Fingerprint	
> 1st key pair for SSH for client1	0a:2f:5b:47:8c:34:f3:01:1b:8e:1a:3b:78:d9:55:0c	⌵

Networks

Displaying 1 item

Network Ports

Security Groups

Key Pair

▼ Available 0

Select one

🔍 Click here for filters. ✕

Configuration

Displaying 0 items

Server Groups

Name	Fingerprint
No items to display.	

Scheduler Hints

Metadata

Displaying 0 items

✕ Cancel

< Back

Next >

🚀 Launch Instance

Image 67: We select a key pair to be used for SSH connections

Details

You can customize your instance after it has launched using the options available here. "Customization Script" is analogous to "User Data" in other systems.



Source

Load Customization Script from a file

No file chosen

Flavor

Customization Script

Content size: 0 bytes of 16.00 KB

Networks

Network Ports

Security Groups

Key Pair

Configuration

Disk Partition

Automatic ▾

Server Groups

Configuration Drive

Scheduler Hints

Metadata

✕ Cancel

< Back

Next >

Launch Instance

Image 68: We have no special configuration script for the instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Select the server group to launch the instance in.

Allocated

Name

Select a server group from the available groups below.

Available 0 Select one

Filter

Name

No available items

✕ Cancel < Back Next > Launch Instance

Image 69: We haven't created any server groups to launch the instance in

- Details
- Source
- Flavor
- Networks
- Network Ports
- Security Groups
- Key Pair
- Configuration
- Server Groups
- Scheduler Hints
- Metadata

This step allows you to add Metadata items to your instance. ?

You can specify resource metadata by moving items from the left column to the right column. In the left column there are metadata definitions from the Glance Metadata Catalog. Use the "Custom" option to add metadata with the key of your choice.

Available Metadata Filter

Custom +

- › Database Software +
- › Runtime Environment +
- › Web Servers +

Existing Metadata Filter

No existing metadata

Click each item to get its description here.

✕ Cancel

< Back

Next >

🔒 Launch Instance

Image 71: We can use many instance metadata for special configuration of the instance

Instances

Instance ID =

Displaying 1 item

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	1st instance with cirros for client1	-	10.0.0.68	m1.tiny	1st key pair for SSH for client1	Build	nova	Block Device Mapping	No State	0 minutes	Associate Floating IP

Displaying 1 item

Image 72: The new instance is being built

Instances

Instance ID =

Displaying 1 item

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	1st instance with cirros for client1	-	10.0.0.68	m1.tiny	1st key pair for SSH for client1	Active	nova	None	Running	0 minutes	Create Snapshot

Displaying 1 item

Image 73: The new instance is displayed at running state

1st instance with cirros for client1

Create Snapshot ▾

Overview Interfaces Log Console Action Log

Name	1st instance with cirros for client1
Description	1st instance with cirros for client1
ID	ac09adc4-c1d2-469d-afb9-b96cba9b2023
Status	Active
Locked	False
Availability Zone	nova
Created	July 21, 2018, 7:21 p.m.
Time Since Created	2 minutes

Specs

Flavor Name	m1.tiny
Flavor ID	1
RAM	512MB
VCPUs	1 VCPU
Disk	1GB

IP Addresses

Private	10.0.0.68
----------------	-----------

Security Groups

default	ALLOW IPv6 to ::/0 ALLOW IPv6 from default ALLOW IPv4 from default ALLOW IPv4 to 0.0.0.0/0
SSH and ICMP (ping) fo...	ALLOW IPv4 icmp from SSH and ICMP (ping) for Linux VMs ALLOW IPv6 to ::/0 ALLOW IPv4 to 0.0.0.0/0 ALLOW IPv4 22/tcp from SSH and ICMP (ping) for Linux VMs ALLOW IPv4 icmp to SSH and ICMP (ping) for Linux VMs

Image 74: An overview of the instance with its private IP address displayed

Metadata

Key Name	1st key pair for SSH for client1
Image	None

Volumes Attached

Attached To 05f09468-9095-4691-bf98-4c94e2153368 on /dev/vda

Image 75: Overview of the instance continued

1st instance with cirros for client1

Create Snapshot ▾

Overview Interfaces Log Console Action Log

Instance Console

If console is not responding to keyboard input: click the grey status bar below. [Click here to show only console](#)
To exit the fullscreen mode, click the browser's back button.

```
Connected (unencrypted) to: QEMU (instance-00000002) Send CtrlAltDel
[ 2.852797] cpuidle: using governor ladder
[ 2.853350] cpuidle: using governor menu
[ 2.854192] EFI Variables Facility v0.08 2004-May-17
[ 2.860831] TCP cubic registered
[ 2.864174] NET: Registered protocol family 10
[ 2.876821] NET: Registered protocol family 17
[ 2.877498] Registering the dns_resolver key type
[ 2.913949] registered taskstats version 1
[ 2.922503] Freeing initrd memory: 3452k freed
[ 3.130813] Magic number: 10:423:395
[ 3.133003] rtc_cmos 00:01: setting system clock to 2018-07-21 19:22:01 UTC (
1532200921)
[ 3.134049] powernow-k8: Processor cpuid 663 not supported
[ 3.137113] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[ 3.137965] EDD information not available.
[ 3.156385] Freeing unused kernel memory: 928k freed
[ 3.187303] Write protecting the kernel read-only data: 12288k
[ 3.213051] Freeing unused kernel memory: 1596k freed
[ 3.256536] Freeing unused kernel memory: 1184k freed

further output written to /dev/ttyS0

login as 'cirros' user. default password: 'cubswin:)', use 'sudo' for root.
1st-instance-with-cirros-for-client1 login:
```

Image 78: The console of the instance

1st instance with cirros for client1

Create Snapshot ▾

Overview Interfaces Log Console Action Log

Displaying 1 item

Request ID	Action	Start Time	User ID	Message
req-71c25964-1855-4f59-a7e0-7ccde00b6cb0	Create	July 21, 2018, 7:21 p.m.	40da007b117b47eaad974bcdda3f46e7	-

Displaying 1 item

Image 79: The action log of the instance

5.6.3 A floating IP allocation

We can allocate a floating IP address to an instance either while it is being built or from the overview screen of it. Another way is from the floating address management screen.

1st instance with cirros for client1

Overview Interfaces Log Console Action Log

Name 1st instance with cirros for client1
Description 1st instance with cirros for client1
ID ac09adc4-c1d2-469d-afb9-b96cba9b2023
Status Active
Locked False
Availability Zone nova
Created July 21, 2018, 7:21 p.m.
Time Since Created 2 minutes

Specs

Flavor Name m1.tiny
Flavor ID 1
RAM 512MB
VCPUs 1 VCPU
Disk 1GB

IP Addresses

Private 10.0.0.68

Security Groups

Create Snapshot

- Associate Floating IP
- Attach Interface
- Detach Interface
- Edit Instance
- Attach Volume
- Detach Volume
- Update Metadata
- Edit Security Groups
- Edit Port Security Groups
- Console
- View Log
- Pause Instance
- Suspend Instance
- Shelve Instance
- Resize Instance
- Lock Instance
- Soft Reboot Instance
- Hard Reboot Instance
- Shut Off Instance
- Rebuild Instance
- Delete Instance

Image 80: We can associate a floating IP address to the instance from the instance overview screen

Manage Floating IP Associations

IP Address *

No floating IP addresses allocated +

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

1st instance with cirros for client1: 10.0.0.68

Associate

Image 81: We add a floating IP address to the instance we have created

Allocate Floating IP



Pool *

Description:

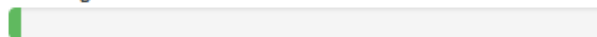
Allocate a floating IP from a given floating IP pool.

Description

Project Quotas

Floating IP

0 of 50 Used



Cancel

Allocate IP

Image 82: We select the public network to be the pool that will give the floating IP address and give a description to it

Manage Floating IP Associations

IP Address *

Select the IP address you wish to associate with the selected instance or port.

Port to be associated *

Associate

Image 83: Here we see the floating IP address and associate it to the instance

1st instance with cirros for client1

Overview Interfaces Log Console Action Log

Name	1st instance with cirros for client1
Description	1st instance with cirros for client1
ID	ac09adc4-c1d2-469d-afb9-b96cba9b2023
Status	Active
Locked	False
Availability Zone	nova
Created	July 21, 2018, 7:21 p.m.
Time Since Created	1 hour

Specs

Flavor Name	m1.tiny
Flavor ID	1
RAM	512MB
VCPUs	1 VCPU
Disk	1GB

IP Addresses

Private	10.0.0.68, 192.168.229.235
----------------	----------------------------

Security Groups

default	ALLOW IPv6 to ::/0 ALLOW IPv6 from default ALLOW IPv4 from default ALLOW IPv4 to 0.0.0.0/0
SSH and ICMP (ping) fo...	ALLOW IPv4 icmp from SSH and ICMP (ping) for Linux VMs ALLOW IPv6 to ::/0 ALLOW IPv4 to 0.0.0.0/0 ALLOW IPv4 22/tcp from SSH and ICMP (ping) for Linux VMs ALLOW IPv4 icmp to SSH and ICMP (ping) for Linux VMs

Image 84: We can see the floating IP address along with the private one in the overview of the instance

Floating IPs

Floating IP Address = ▾

Filter

🔗 Allocate IP To Project

🔗 Release Floating IPs

Displaying 1 item

<input type="checkbox"/>	IP Address	Description	Mapped Fixed IP Address	Pool	Status	Actions
<input type="checkbox"/>	192.168.229.235	1st IP for 1st instance with cirros for client1	1st instance with cirros for client1 10.0.0.68	public	Active	Disassociate ▾

Displaying 1 item

Image 85: The screen with the floating IPs of the project

Network Topology with the first instance of client1 project:

Project / Network / Network Topology

Network Topology

Topology Graph

Small Normal

The diagram illustrates a network topology with two vertical bars representing networks. The left bar is blue and labeled 'public'. The right bar is orange and labeled 'private' with the IP address '10.0.0.64/26' below it. A router icon connects the two networks. An instance icon is connected to the private network. A tooltip for the instance shows:

- 1st instance with cirros for client1
- ID ac09adc4-c1d2-469d-afb9-b96cba9b2023
- STATUS Active
- » View Instance Details
- » Open Console

Launch Instance + Create Network + Create Router

Image 86: The network topology of the project with the instance shown

6 Conclusions

The aim of this paper is to investigate the possibility of improving the Information Technology services in C.E.R.T.H. As with any research organization, there are constantly changing requirements in a wide range depending on the projects that are being implemented at any moment. The usual practice is to purchase equipment that is used for a short time and then under-operates or is removed.

A solution to this problem can be given by Cloud Computing and in particular the Infrastructure as a Service (IaaS). It can provide the necessary resources according to demand, with an easy way for users without dependence on administrators who can deal with more basic and essential issues. This enables more efficient use of resources and reduced equipment and management costs.

The characteristics of the IaaS sought and tested are continuous availability, elasticity, cost reduction and energy savings, security and possible vendor lock in.

The most appropriate type of cloud for this case is also investigated. That is, if a private, public, community or hybrid cloud is preferable.

Availability is achieved by coexisting applications - copied or transferred - to more than one point at a time. Separate points can be server clusters at the same data center or even in different ones in different geographic areas, ensuring seamless operation even in the event of major disasters. Each resource must have an alternative, whether it concerns a server and a network device or software so that applications are not interrupted.

Elasticity is the fluctuation of resources according to the load, which leads to lower operating costs and customer expenses. It is distinguished in horizontal and vertical. In horizontal, the number of virtual machines of an application fluctuates depending on the load, while in vertical the virtual machine's resources (power, memory, etc.) fluctuate. This is achieved either by time schedule or user-defined thresholds of use.

Using virtualization technology we have the ability to host many virtual servers on one physical. This helps us get closer to 100% use of the available resources. So this way we save energy. The use of renewable energy sources also helps reducing operating costs and consequently the cost to the customer.

In terms of security, the issue is to avoid the negatives of multi-tenancy, i.e. the multiple lease of the same resources to many customers, by isolating virtual machines and applications between them and hiding the location of the data. At the level of hardware and software management, all international safety standards must be followed and infrastructure must follow all international organizations' compliance. For data security, encryption at all levels and secure identity management are necessary.

To avoid the entrapment of the customer to a provider, the clouds must use open standards to enable interoperability and the transfer of virtual machines between them, helping users to choose the right one and the service they desire at any given time.

Next, a parallel presentation of the IaaS service of 3 commercial clouds (Amazon's Elastic Compute Cloud (EC2), Google Compute Engine and Microsoft Azure), as well as the one of OpenStack, the most popular private / proprietary cloud development platform, is done.

It is noted that competition has led to a similar product by all tested solutions, despite companies' claims for pioneering solutions, etc. All of them provide many types of virtual machines with different capabilities. There are many types of storage, the temporary one built-in on the VM, the permanent one, the shared one, etc. Virtual networks can be implemented, with ephemeral and permanent IP addresses, firewalls, routers and whatever else is required. Indicative of their homogeneity is that they are continuously implementing non-originally deliverables such as the provision of SSDs by Amazon and of storage for archiving by Microsoft.

In terms of elasticity, the four implementations provide only horizontal. Vertical exists but it is not real-time dynamic, it requires shutdown and VM type change.

The effort for security and privacy is a given. All known practices and compliance with all national and international standards are followed. This does not guarantee, of course, that any cloud is immune. Like any IT infrastructure, there are attacks and a history of them for each cloud.

Extremely high is the availability of at least the 3 commercial clouds, since OpenStack depends on the implementation. There are different regions and / or

availability zones with independent data centers where applications can be hosted to ensure their operation.

All of the clouds examined support modern virtualization technologies, achieving a significant reduction in operating costs. Both Amazon and Google are approaching 100% of renewable energy use. But Microsoft has not yet reached 50%. This can repel those who value ecological issues, but it can also mean increased prices, both because of the cost of purchasing energy and that of pollutant certificates.

Amazon uses its own interface standards and APIs that are currently de facto standards but can lead to customer entrapment at any time. Google uses open standards, but there is an issue with exporting VMs for use on another infrastructure. Although RAW is supported, there are problems. The most open cloud is Azure, which is from the beginning implemented for this purpose. However, there is a possibility the entrapment to result in PaaS and SaaS, through the .NET development platform, the Office 365 etc.

The OpenStack simulation has shown that we can approach 100% of the available resources of a physical server by adding virtual machines, depending on their applications and requirements over time. It is also possible to completely isolate the virtual machines and the network of a tenant-client from the rest. No one has access to, nor knows the resources of, anyone else unless it is desirable (especially in the case of private clouds), so it can be arranged by the administrator.

According to the above, the proposal for C.E.R.T.H. is to implement a private cloud based on OpenStack initially, to take advantage of the benefits of technology. Then it can be converted to hybrid if MyCERTH is transferred to Microsoft Azure, as it is based on .NET Framework and SQL Server. At the same time other clouds can be considered, depending on the cost and benefits of the season, since they are constantly changing.

The implementation of cloud from GRNET currently used to a small extent is not sufficient, since there are things that have not yet been implemented (e.g. archive storage is in alpha stage) and there are restrictions as resources are insufficient (e.g. only 2 VMs per user in Okeanos service are provided). Later, however, the solution may be a community cloud for all institutions that are engaged in research based on GRNET.

An object of future research may be the implementation of a private cloud based on OpenStack. Also separate fields for research are all features that have been examined here to a lesser extent, namely elasticity, vendor lock in and of course security. Interesting fields also are the PaaS and SaaS services and perhaps later the GRNET cloud, when it will reach higher levels of implementation.

7 References

Articles

Maricela-Georgiana Avram (Olaru) (2013), “Advantages and challenges of adopting cloud computing from an enterprise perspective”, 2013 The 7th International Conference Interdisciplinarity in Engineering (INTER-ENG 2013), ELSEVIER, ScienceDirect 2014

Tiago Oliveira, Manoj Thomas, Mariana Espadanal (2014), “Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors”, ELSEVIER, ScienceDirect 2014

Jiunn-Woei Lian, David C. Yen, Yen-Ting Wang (2013), “An exploratory study to understand the critical factors affecting the decision to adopt cloud computing in Taiwan hospital”, ELSEVIER, ScienceDirect 2014

Christos Stergiou, Kostas E. Psannis, Brij B. Gupta and Yutaka Ishibashi (2018), “Security, Privacy & Efficiency of Sustainable Cloud Computing for Big Data & IoT”, ELSEVIER, ScienceDirect 2018

Christos Stergiou, Kostas E. Psannis, Theofanis Xifilidis, Andreas P. Plageras and Brij B. Gupta (2018), “Security and Privacy of Big Data for Social Networking Services in Cloud”, IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)

Kostas E. Psannis, Christos Stergiou and B. B. Gupta (2018), “Advanced Media-based Smart Big Data on Intelligent Cloud Systems”, IEEE Transactions on Sustainable Computing (Early Access)

Christos Stergiou and Kostas E. Psannis (2017), “Efficient and secure BIG data delivery in Cloud Computing”, SpringerLink, Multimedia Tools and Applications, November 2017

Ahmad M. Manasrah and Hanan Ba Ali (2018), “Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing”, Hindawi, Wireless Communications and Mobile Computing, January 2018

José I. Benedetto, Guillermo Valenzuela, Pablo Sanabria, Andrés Neyem, Jaime Navón and Christian Poellabauer (2018), “MobiCOP: A Scalable and Reliable Mobile Code

Offloading Solution”, Hindawi, Wireless Communications and Mobile Computing, January 2018

Xun Wu (2018), “Context-Aware Cloud Service Selection Model for Mobile Cloud Computing Environments”, Hindawi, Wireless Communications and Mobile Computing, March 2018

Christos Stergiou, Kostas E. Psannis, Byung-Gyu Kimb and Brij Gupta (2018), “Secure integration of IoT and Cloud Computing”, ELSEVIER, ScienceDirect 2018

Christos Stergiou and Kostas E. Psannis (2016), “Recent advances delivered by mobile cloud computing and Internet of Things for Big data applications: A Survey”, Wiley Online Library 2016

Vasileios A. Memos and Kostas E. Psannis (2014), “A New Methodology Based on Cloud Computing for Efficient Virus Detection”, SpringerLink, New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering

K.E. Psannis*, S. Xinogalos and A. Sifaleras (2014), “Convergence of Internet of things and mobile cloud computing”, Taylor&FrancisOnline, Journal Systems Science & Control Engineering

Victor Chang, Yen-Hung Kuo and Muthu Ramachandran (2015), “Cloud computing adoption framework: A security framework for business clouds”, ELSEVIER, ScienceDirect 2016

Rostyslav Zabolotnyi, Philipp Leitner, Waldemar Hummer and Schahram Dustdar (2015), “JCloudScale: Closing the Gap Between IaaS and PaaS”, Cornell University Library

Carlos Mera-Gomez, Francisco Ramirez, Rami Bahsoon and Rajkumar Buyya (2017), “A Debt-Aware Learning Approach for Resource Adaptations in Cloud Elasticity Management”, SpringerLink, International Conference on Service-Oriented Computing

Victor Gonzalez Chamorro, Carlos Nunez Castillo and Fabio Lopez-Pires (2016), “An Elastic VoIP Solution based on OpenStack”, 2016 International Conference on Information Systems Engineering (ICISE), IEEE

Kaveh Razavi Gerrit Van Der Kolk and Thilo Kielmann (2015), “Prebaked μ VMs: Scalable, Instant VM Startup for IaaS Clouds”, 2015 IEEE 35th International Conference on Distributed Computing Systems, IEEE

Dan Gonzales, Jeremy M. Kaplan, Evan Saltzman, Zev Winkelman, and Dulani Woods (2015), “Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds”, IEEE Transactions on Cloud Computing, July-September 2017

Amir Teshome, Louis Rilling and Christine Morin (2018), ” Verification for security monitoring SLAs in IaaS clouds: The example of a network IDS”, NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, 2018

Jinho Seol, Seongwook Jin, Daewoo Lee, Jaehyuk Huh, and Seungryoul Maeng (2016), “A Trusted IaaS Environment with Hardware Security Module”, IEEE Transactions on Services Computing, May-June 2016

Nidal Hassan Hussein and Ahmed Khalid (2016), “A survey of Cloud Computing Security challenges and solutions”, International Journal of Computer Science and Information Security (IJCSIS), January 2016

Flora Amato, Francesco Moscato, Vincenzo Moscato and Francesco Colace (2017), “Improving security in cloud by formal modeling of IaaS resources”, ELSEVIER, ScienceDirect 2017

Fei Teng, Lei Yu, Tianrui Li, Danting Deng and Frédéric Magoulès (2016), “Energy efficiency of VM consolidation in IaaS clouds”, SpringerLink, The Journal of Supercomputing, February 2017

A. Kertesz, J. D. Dombi and A. Benyi (2015), “A Pliant-based Virtual Machine Scheduling Solution to Improve the Energy Efficiency of IaaS Clouds”, SpringerLink, Journal of Grid Computing, 2015

Syed Hamid Hussain Madni, Muhammad Shafie Abd Latiff, Yahaya Coulibaly, and Shafi’i Muhammad Abdulhamid (2016), “Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities”, ELSEVIER, ScienceDirect 2016

Sangdo Lee Hyoungyill Park and Yongtae Shin (2012), “Cloud Computing Availability: Multi-clouds for Big Data Service”, SpringerLink, International Conference on Hybrid Information Technology, 2012

Panagiotis Kokkinos, Dimitris Kalogeras, Anna Levin and Emmanouel Varvarigos (2016), “Survey: Live Migration and Disaster Recovery over Long-Distance Networks”, ACM Digital Library, Journal ACM Computing Surveys (CSUR), 2016

L. Tomás, P. Kokkinos, V. Anagnostopoulos, O. Feder, D. Kyriazis, K. Meth, E. Varvarigos and T. Varvarigou (2017). “Disaster Recovery Layer for Distributed OpenStack Deployments”, IEEE IEEE Transactions on Cloud Computing (Early Access), August 2017

Aqeel Sahi, David Lai and Yan Li (2016), “Security and privacy preserving approaches in the eHealth clouds with Disaster recovery plan”, ELSEVIER, ScienceDirect 2016

Long Wang, Richard E Harper, Ruchi Mahindru, and Harigovind V Ramasamy (2016), “Disaster Recovery for Cloud-Hosted Enterprise Applications”, IEEE 9th International Conference on Cloud Computing, 2016

Justice Opara-Martins, Reza Sahandi and Feng Tian (2016), “Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective”, SpringerLink, Journal of Cloud Computing, December 2016

Eslam G. AbdAllah, Mohammad Zulkernine, Yuan Xiang Gu and Clifford Liem (2017), “TRUST-CAP: A Trust Model for Cloud-based Applications”, 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)

Deepak Puthal, B. P. S. Sahoo, Sambit Mishra, and Satyabrata Swain (2015), “Cloud Computing Features, Issues and Challenges: A Big Picture”, 2015 International Conference on Computational Intelligence & Networks

Nikolas Roman Herbst, Samuel Kounev, Andreas Weber and Henning Groenda (2015), “BUNGEE: An Elasticity Benchmark for Self-Adaptive IaaS Cloud Environments”, 2015 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems

Guilherme Galante, Luis Carlos Erpen De Bona, Antonio Roberto Mury, Bruno Schulze, Rodrigo da Rosa Righi (2016), “An Analysis of Public Clouds Elasticity in the Execution of Scientific Applications: a Survey”, Springer Science+Business Media Dordrecht 2016

Yazhou Hu, Bo Deng, Yu Yang, Dongxia Wang (2016), “Elasticity Evaluation of IaaS Cloud Based on Mixed Workloads”, 2016 15th International Symposium on Parallel and Distributed Computing

Kai Hwang, Xiaoying Bai, Yue Shi, Muyang Li, Wen-Guang Chen and Yongwei Wu (2015), “Cloud Performance Modeling and Benchmark Evaluation of Elastic Scaling Strategies”, IEEE Transactions on Parallel and Distributed Systems

Mohamed Al Morsy, John Grundy and Ingo Müller (2016), “An Analysis of the Cloud Computing Security Problem”, Proceedings of the APSEC 2010 Cloud Workshop, Sydney, Australia, 30th Nov 2010

Amani S. Ibrahim, James Hamlyn-Harris, John Grundy (2016), “Emerging Security Challenges of Cloud Virtual Infrastructure”, Proceedings of the APSEC 2010 Cloud Workshop, Sydney, Australia, 30th Nov 2010

Soumya Ranjan Jena, V. Vijayaraja and Aditya Kumar Sahu, (2016), “Performance Evaluation of Energy Efficient Power Models for Digital Cloud”, Indian Journal of Science and Technology, Vol 9(48), December 2016

Ismael Cuadrado-Cordero, Anne-Cecile Orgerie, Jean-Marc Menaud (2017), “Comparative Experimental Analysis of the Quality-of-Service and Energy-Efficiency of VMs and Containers' Consolidation for Cloud Applications”, SoftCOM: International Conference on Software, Telecommunications and Computer Networks, Sep 2017, Split, Croatia

Rahul Ghosh, Francesco Longo, Flavio Frattini, Stefano Russo, and Kishor S. Trivedi (2014), “Scalable Analytics for IaaS Cloud Availability”, IEEE Transactions on Cloud Computing, January 2014

Ravi Jhavar and Vincenzo Piuri (2012), “Fault Tolerance Management in IaaS Clouds”, Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference

Álvaro López García, Enol Fernández del Castillo, Pablo Orviz Fernández (2016), “Standards for enabling heterogeneous IaaS cloud federations”, Institute of Physics of Cantabria, Spanish National Research Council

Anton Beloglazov and Rajkumar Buyya (2014), “OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds”, Wiley Online Library (wileyonlinelibrary.com)

Rajyalakshmi Marathu, Divya K Konoor and Prashanth Reddy (2016), “Secure OpenStack Cloud with Bandit”, 2016 IEEE International Conference on Cloud Computin in Emerging Markets

David W. Chadwick, Kristy Siu, Craig Lee, Yann Fouillat, Damien Germonville (2013), “Adding Federated Identity Management to OpenStack”, Springerlink.com

Web sites

CERTH At a glance:

<http://certh.gr/5B4D1A98.en.aspx>

Amazon EC2 Storage:

<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/Storage.html>

Amazon Glacier documentation:

<https://aws.amazon.com/documentation/glacier/>

AWS Regions and Availability Zones:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html>

AWS Elastic Load Balancing:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/what-is-load-balancing.html>

Amazon EC2 Frequently Asked Questions (FAQs):

<https://aws.amazon.com/ec2/faqs/>

Amazon Elastic Compute Cloud documentation:

<https://aws.amazon.com/documentation/ec2/>

AWS documentation:

<https://aws.amazon.com/documentation/>

AWS News Blog, “Cloud Computing, Server Utilization, & the Environment”:

<https://aws.amazon.com/blogs/aws/cloud-computing-server-utilization-the-environment/>

AWS & Sustainability:

<https://aws.amazon.com/about-aws/sustainability/>

Amazon, “Energy and Environment”:

<https://www.amazon.com/p/feature/gkkwdp34z5ou7ug>

Amazon Web Services, “Introduction to AWS Security”. July 2015:

https://d1.awsstatic.com/whitepapers/Security/Intro_to_AWS_Security.pdf

Amazon Web Services, “Overview of Security Processes”, June 2016:

<http://d0.awsstatic.com/whitepapers/Security/AWS%20Security%20Whitepaper.pdf>

Amazon Web Services, “Amazon Web Services: Risk and Compliance”, May 2017:

https://d1.awsstatic.com/whitepapers/compliance/AWS_Risk_and_Compliance_Whitepaper.pdf

Google Compute Engine Documentation:

<https://cloud.google.com/compute/docs/>

Google Cloud Platform, “Applying Sizing Recommendations for VM Instances”:

<https://cloud.google.com/compute/docs/instances/apply-sizing-recommendations-for-instances>

Google Cloud Platform, “Google Cloud and the Environment”:

<https://cloud.google.com/environment/>

Google Cloud Platform, “Google Cloud Platform Security”:

<https://cloud.google.com/security/>

Google Cloud Platform Blog, “Titan in Depth: Security in plaintext”, August 2017:

<https://cloudplatform.googleblog.com/2017/08/Titan-in-depth-security-in-plaintext.html>

Google Cloud Platform, “Encryption in Transit in Google Cloud”:

<https://cloud.google.com/security/encryption-in-transit/>

Google Cloud Platform, “Encryption at Rest in Google Cloud Platform”:

<https://cloud.google.com/security/encryption-at-rest/default-encryption/>

Google Cloud Platform, “Google Cloud Platform Security: Google Cloud Platform meets rigorous privacy and compliance standards that test for data safety, privacy, and security”:

<https://cloud.google.com/security/compliance>

Microsoft Azure, “Virtual Machines”:

<https://azure.microsoft.com/en-us/services/virtual-machines/>

Microsoft Azure, “Linux Virtual Machines”:

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/>

Microsoft Azure, “Create a Windows VM from a specialized disk using PowerShell”, January 2017:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/create-vm-specialized>

Kunal Chandratre, “Microsoft Azure platform Demystified - Part One & Two”, February 2016:

<http://www.dotnetcurry.com/windows-azure/1299/microsoft-azure-platform-services-overview>

Microsoft Azure, “Sizes for Windows virtual machines in Azure”, August 2017:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes>

Google Cloud Platform, “Google Cloud Platform for Azure Professionals: Compute”, July 2017:

<https://cloud.google.com/docs/compare/azure/compute>

Microsoft Azure, “Automatically scale virtual machines in Azure” August 2017:

<https://docs.microsoft.com/en-us/azure/virtual-machines/windows/autoscale>

Microsoft Azure, “Manage Azure Virtual Networks and Linux Virtual Machines with the Azure CLI”, May 2017:

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/tutorial-virtual-network>

Google Cloud Platform, “Google Cloud Platform for Azure Professionals: Networking”, July 2017:

<https://cloud.google.com/docs/compare/azure/networking>

Microsoft Azure, “Azure Regions”:

<https://azure.microsoft.com/en-us/regions/>

Microsoft Azure, “Azure and Linux”, November 2017:

<https://docs.microsoft.com/en-us/azure/virtual-machines/linux/overview?toc=%2fazure%2fvirtual-machines%2flinux%2fclassic%2ftoc.json>

Microsoft Azure, “Azure Blob Storage: Hot, cool, and archive storage tiers”, December 2017:

<https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers>

Microsoft Azure, “Introduction to Microsoft Azure Storage”, November 2017:

<https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction>

Microsoft Azure, “What is Azure Policy?” , January 2018:

<https://docs.microsoft.com/en-us/azure/azure-policy/azure-policy-introduction>

Microsoft Azure, “Cost Management”:

<https://azure.microsoft.com/en-us/services/cost-management/>

Microsoft Azure Blog Announcements, “New Azure management and cost savings capabilities”, December 2017:

<https://azure.microsoft.com/en-us/blog/new-azure-management-and-cost-savings-capabilities/>

Microsoft Azure, “Azure Advisor:

<https://azure.microsoft.com/en-us/services/advisor/>

Microsoft Azure documentation, “Introduction to Azure Advisor”, November 2016:

<https://docs.microsoft.com/en-us/azure/advisor/advisor-overview>

Microsoft Azure Blog Updates, “Achieve better savings with best-in-class cost management on Azure”, October 2017:

<https://azure.microsoft.com/en-us/blog/achieve-better-savings-with-best-in-class-cost-management-on-azure/>

Brad Smith - President and Chief Legal Officer, The Official Microsoft Blog, “Greener datacenters for a brighter future: Microsoft’s commitment to renewable energy”, May 2016:

<https://blogs.microsoft.com/on-the-issues/2016/05/19/greener-datacenters-brighter-future-microsofts-commitment-renewable-energy/>

Microsoft Azure documentation, “Getting started with Microsoft Azure security”, November 2017:

<https://docs.microsoft.com/en-us/azure/security/azure-security-getting-started>

Wikipedia, “OpenStack”:

<https://en.wikipedia.org/wiki/OpenStack>

redhat, “Understanding OpenStack”:

<https://www.redhat.com/en/topics/openstack#>

NIST, “NIST Cloud Computing Program - NCCP”:

<https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp>

OpenStack Documentation, “Feature Classification”, January 2018:

<https://docs.openstack.org/nova/latest/user/feature-classification.html>

webopedia, “OpenStack Nova”:

<https://www.webopedia.com/TERM/O/openstack-nova.html>

github.com, “OpenStack Nova”:

<https://github.com/openstack/nova>

OpenStack Documentation, “OpenStack Compute (nova)”, January 2018:

<https://docs.openstack.org/nova/latest/>

OpenStack Wiki, “Neutron”:

<https://wiki.openstack.org/wiki/Neutron>

OpenStack Documentation, “Networking (neutron) concepts”, November 2017:

<https://docs.openstack.org/mitaka/install-guide-ubuntu/neutron-concepts.html>

OpenStack Documentation, “Storage concepts”, January 2017:

<https://docs.openstack.org/arch-design/design-storage/design-storage-concepts.html>

opensource.com, “What is OpenStack? “:

<https://opensource.com/resources/what-is-openstack>

OpenStack Documentation, “Welcome to Glance’s documentation!”, August 2017:

<https://docs.openstack.org/glance/pike/>

OpenStack Wiki, “Watcher”:

<https://wiki.openstack.org/wiki/Watcher>

openstack.org Documentation, “Energy Saving Strategy”:

<http://specs.openstack.org/openstack/watcher-specs/specs/pike/implemented/energy-saving-strategy.html>

openstack-neat.org, “OpenStack Neat”:

<http://openstack-neat.org/>

OpenStack Documentation, “OpenStack Security Guide”:

<https://docs.openstack.org/security-guide/>

openstack.org Documentation, “Securing OpenStack Clouds”:

<https://www.openstack.org/assets/securing-openstack-clouds/OpenStack-SecurityBriefletteronline.pdf>

OpenStack Documentation, “Compliance”:

<https://docs.openstack.org/security-guide/compliance.html>

8 Appentices

Standards and Organizations:

	ΣΥΝΤΟΜΟΓΡΑΦΙΑ	URL
Application Programming Interface	API	https://en.wikipedia.org/wiki/Application_programming_interface
Hyper Transfer Protocol Secure	HTTPS	https://en.wikipedia.org/wiki/HTTPS
Simple Object Access Protocol	SOAP	https://en.wikipedia.org/wiki/SOAP
Representational state transfer	REST	https://en.wikipedia.org/wiki/Representational_state_transfer
Remote procedure call	RPC	https://en.wikipedia.org/wiki/Remote_procedure_call
Virtual Private Network	VPN	https://en.wikipedia.org/wiki/Virtual_private_network
File Transfer Protocol	FTP	https://en.wikipedia.org/wiki/File_Transfer_Protocol
Service Provisioning Markup Language	SPML	https://en.wikipedia.org/wiki/Service_Provisioning_Markup_Language
Security Assertion Markup Language	SAML	https://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
OAuth	OAuth	https://en.wikipedia.org/wiki/OAuth
eXtensible Access Control Markup Language	XACML	https://en.wikipedia.org/wiki/XACML
DNS server		https://en.wikipedia.org/wiki/Name_server

Open Grid Forum	OGF	https://en.wikipedia.org/wiki/Open_Grid_Forum https://www.ogf.org/ogf/doku.php
Open Cloud Computer Interface	OCCI	http://occi-wg.org/ https://en.wikipedia.org/wiki/Open_Cloud_Computing_Interface
Distributed Management Task Force	DMTF	http://www.dmtf.org/ https://en.wikipedia.org/wiki/Distributed_Management_Task_Force
Cloud Infrastructure Management Interface	CIMI	https://en.wikipedia.org/wiki/Cloud_Infrastructure_Management_Interface
Organization for the Advancement of Structured Information Standards	OASIS	https://www.oasis-open.org/ https://en.wikipedia.org/wiki/OASIS_(organization)
Topology and Orchestration Specification for Cloud Applications	TOSCA	https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca https://en.wikipedia.org/wiki/OASIS_TOSCA
Storage Networking Industry Association	SNIA	https://www.snia.org/ https://en.wikipedia.org/wiki/Storage_Networking_Industry_Association
Cloud Data Management Interface	CDMI	https://www.snia.org/cdmi https://en.wikipedia.org/wiki/Cloud_Data_Management_Interface
Open Virtualization Format	OVF	https://www.dmtf.org/standards/ovf https://en.wikipedia.org/wiki/Open_Virtualization_Format

Grid Security Infrastructure	GSI	https://en.wikipedia.org/wiki/Grid_Security_Infrastructure
X.509 public key certificate		https://en.wikipedia.org/wiki/X.509
SHIBBOLETH		https://www.shibboleth.net/ https://en.wikipedia.org/wiki/Shibboleth_(Shibboleth_Consortium)
Grid Laboratory Uniform Environment	GLUE	https://en.wikipedia.org/wiki/Open_Grid_Forum
Usage Record (UR) 2.0	UR 2.0	https://en.wikipedia.org/wiki/Open_Grid_Forum
Dynamic Host Configuration Protocol	DHCP	https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
SSL/TLS	Secure Sockets Layer/Transport Layer Security	https://en.wikipedia.org/wiki/Transport_Layer_Security
SSH	Secure Shell	https://www.ssh.com/ssh/protocol/ https://en.wikipedia.org/wiki/Secure_Shell
SMB	Server Message Block	https://en.wikipedia.org/wiki/Server_Message_Block
TDE/CLE	Table Level Encryption/Column Level Encryption	https://en.wikipedia.org/wiki/Database_encryption
IPsec	Internet Protocol Security	https://en.wikipedia.org/wiki/IPsec
AES	Advanced Encryption Standard	https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
SSTP	Secure Socket Tunneling Protocol	https://en.wikipedia.org/wiki/Secure_Socket_Tunneling_Protocol

SDN	Software Defined Networking	https://en.wikipedia.org/wiki/Software-defined_networking
NFS	Network File System	https://en.wikipedia.org/wiki/Network_File_System
CIFS	Common Internet File System	https://technet.microsoft.com/en-us/library/cc939973.aspx
GlusterFS	Gluster File System	https://www.gluster.org/ https://en.wikipedia.org/wiki/Gluster
HDFS	Hadoop Distributed File System	https://en.wikipedia.org/wiki/Apache_Hadoop
LDAP	Lightweight Directory Access Protocol	https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol