Polytechnique Montréal

Escola Tècnica Superior d'Enginyeria de Telecomunicacions de Barcelona

Universitat Politècnica de Catalunya

Security in cloud computing

DEGREE FINAL WORK

Degree in Telematics Engineering

Author: Oriol Moreno Martín

Tutor: Martine Bellaïche

Jose A. Lázaro

Course 2015-2016

To my friends and family, always there when I needed them.
To the people that motivated me to do better and not settle when things got difficult
To Marc Ubiñana and Eric Navarro, who were also working on their project, as we motivated each other and remembered our long journey
To all the persons that I care about, and made me who I am

Acknowledgements

This words would not have been written, in this context, if not for prof. Martine Bellaïche. I want to thank her for accepting me as his student for this 5 months. It has been a pleasure working under her tutelage and learning new concepts and technologies previously unknown to me.

She was welcoming and cared for my work, helped me when needed, and tried to give me assignments I could deliver, knowing my previously inexistent research background. For all this, and the whole Canadian experience she has provided by accepting me, I will be forever grateful.

I also want to express my gratitude to Adel and Talal. Being Phd students under Martine's tutelage, their only obligation towards me was sharing the office where we work, but they asked and helped whenever they could, and the overall stay in their lab was enriching.

Finally, I want to thank prof. José A. Lázaro. From the first day he assured me that going to Canada would be difficult, but that with some work it would be possible. I did that work, and as he had to endure meeting with me more times than the average student, at the end I reached my goal. As the project is ending, I can say that he cared for my work and my situation more than most co-directors of TFG may do, and that it was a good decision to choose him for that role.

Resum

Cloud Computing és un paradigma recent per proporcionar serveis via Internet. Negocis creixen dràsticament gràcies a aquest. Investigadors focalitzen el seu treball en aquest.

L'accés a recursos d'IT flexibles i de baix preu, sota demanda, evita als usuaris planejar per avançat l'aprovisionament, i a empreses estalviar diners que d'altre manera haurien estat invertits en infraestructures.

En aquesta tesi es presenta un estudi del Cloud Computing, els simuladors creats pel seu estudi, i els aspectes de seguretat de la tecnologia.

Paraules clau: cloud computing, seguretat, aws, simulació

Resumen

Cloud Computing es un paradigma reciente para proporcionar servicios via Internet. Negocios crecen drásticamente gracias a este. Investigadores focalizan su trabajo en este.

El acceso a recursos de IT flexibles y de bajo precio, bajo demanda, evita a los usuarios planificar por avanzado el aprovisionamiento, y a empresas ahorrar dinero que de otra forma habría sido invertido en infraestructuras.

En esta tesi se presenta un estudio del Cloud Computing, los simuladores creados para su estudio, y los aspectos de seguridad de la tecnología.

Palabras clave: cloud computing, seguridad, aws, simulación

Abstract

Cloud Computing is a recent paradigm to deliver services over Internet. Businesses grow drastically because of it. Researchers focus their work on it.

The rapid access to flexible and low cost IT resources on an on-demand fashion, allows the users to avoid planning ahead for provisioning, and enterprises to save money that otherwise would have been invested in infrastructures.

In this thesis is presented a study of Cloud Computing, the simulators created for its study, and the security aspects of the technology.

Key words: cloud computing, security, aws, simulation

Contents

Dedication ii

Acknowledgements iii

Contents v

List of Figures vi

List of Tables vi

1 Introduction 1

1.1 Objectives . . . 1

1.2 Structure . . . 1

2 AWS (Amazon) and Azure (Microsoft) 2

3 CloudFlare 4

4 Cloud Simulators 6

4.1 Detailed energy consumption (Green Cloud) . . . 8

4.2 VM migration in Power Aware Data centers (CloudSim) . . . 9

5 Security: challenges and solutions 14

5.1 Communication . . . 14

5.2 Virtualization . . . 17

5.3 Data/storage . . . 23

5.4 Cloud applications and API . . . 26

5.5 Identity management and access control . . . 28

5.6 Contractual and legal aspects . . . 30

6 Deploying a secure app on AWS 32

6.1 Security basics: IAM Groups/Users, Key Pairs and VPC . . . 32

6.2 Application Server: Security Group, IAM Role and EC2 . . . 34

6.3 Database Server: Security Group, RDS . . . 35

6.4 Deployment of the app: connecting and configuring . . . 35

6.5 Enhancement of the architecture: Auto Scaling, Load Balancing and Domain Name . . . 35

7 Conclusions 38

Bibliography 39

List of Figures

4.1 Three-tier high-speed data center 9
4.2 Simulations results for NonPowerAware (above) and DVFS+DNS (below) 10

6.1 Example architecture for a webapp on AWS 33
6.2 Wordpress website on instance 52.38.218.71 36
6.3 Piwigo webapp on instance 52.38.218.71 36

List of Tables

4.1 Comparison of simulators 8
4.2 Simulation results when migrating VMs for different detection techniques with MMT selection policy 12
4.3 Simulation results when migrating VMs for different selection policies with IQR detection technique 12

1 Introduction

Cloud Computing is one of the emerging technologies to provide services over the Internet. As such, it presents great benefits but requires further investigation to mitigate its flaws. The main one is security, making the user think twice to employ this technology.

1.1 Objectives

The main objective is to acquire a view in depth of Cloud Computing and its security. Analyzing its flaws and familiarizing with its functionalities.

Researching about the simulation of Cloud Networks, as well as methodology to consume less energy, is desired too.

Deploying an app on a Cloud Network like Amazon Web Service will allow to visualize some of the concepts learned.

1.2 Structure

The project will first provide an overview of the Cloud Computing technology, focused a little bit on security, and CloudFlare, a Content Delivery Network widely used for its capabilities.

After that, it will detail the capacities of different Cloud Simulators, making comparisons between them and running some simulations to verify their functionalities.

Finally, the full focus will be on the main security challenges and solutions found in research, and the deployment of an app in the AWS infrastructure with basic security configurations.

2

AWS (Amazon) and Azure (Microsoft)

AWS and Azure are cloud provider platforms that offer computing and storage resources for their clients, with the flexibility of acquiring and releasing the quantity of this resources depending on the demand, and charging the costumer just for the capacity used [1]. Security in both platforms, being Amazon IaaS, and Azure Paas, is shared between the infrastructure provider and the client. So both cloud platforms have the responsibility to protect and secure their network devices, infrastructure, and provide security features to the costumer. And the client is responsible for everything put on the cloud.

Amazon Web Services (AWS) provides a modular network composed of Regions and Availability Zones, that helps you withstand attacks or errors in a particular datacenter, in order to always have the data and resources available [2]. The connection to the cloud is made through monitored access points that allow HTTP and HTTPS connections using SSL for communications security. If the client wants better security, there is the option of establishing a Virtual Private Cloud (VPC) as a subnet within AWS, adding a VPN between the cloud and the client's datacenter for secure communication.

Traditional network security issues like DDoS attacks, Man in the Middle attacks, IP Spoofing, Port Scanning and Packet sniffing by other tenants, are avoided with mitigation techniques of AWS, but since the client has almost total control of the network, he is responsible for the security of the system when opening ports or logging and working without safe protocols like SSL.

Authentication in the AWS account can be secure using credentials like Passwords, Access Keys, Key Pairs and X.509 Certificates, and it also provides the option of asking for a Multi-Factor Authentication (MFA) that increases the difficulty of unauthorized use.

Confidentiality, such as instance isolation and secure data are acquired through the hypervisor and the firewall, that make each instance as if it was in its own physical server with its own resources. Moreover, the firewall is first set in deny-all mode and the client has the responsibility to open each port or establish each rule in order to protect the instance from a security breach due to unauthorized access or a network attack.

Azure provides a platform already configured, for clients, mostly developers, to build or run their own cloud applications.

Like AWS, all communications with the client's datacenter or the Internet have to pass through a firewall that by default declines all input data, except for remote management ports.

To secure communications between internal VM or inbound data from the exterior, it is recommended to use Virtual Networks and encrypt the data with SSL or other application-level encryption techniques [3]. Communication across subscriptions should be done by configuring the VMs to exchange data via public virtual IP addresses.

Normally, the client requires to interconnect its own datacenter and the Azure Virtual Network. To protect this channel, it is recommended to use a Virtual Network Gateway. VNG establishes an IP tunnel to route traffic between both endpoints and depending on the situation of this endpoints, it will be encrypted with AES-256, or the data will just be sent over an MPLS network, which offers improved security.

As in AWS, Azure ensures protecting the platform against DDoS attacks from both the outside and the inside, and to isolate each VM from the others in a server by deploying Network Security Groups providing full control over traffic.

Reviewing all the features of both platforms, we can say that they provide authentication, confidentiality, integrity, availability and isolation, all key aspects to the security of Cloud Computing.

3

CloudFlare

CloudFlare is a CDN (Content Delivery Network) that acts as a reverse proxy for websites, resulting in faster page load times and best performance [4]. As part of CloudFlare community the website will also avoid threats and limit the wasting of bandwidth and resources, originated by bots and crawlers. As its setup only needs a few changes in the domain's DNS settings, the use of the network can be activated or turned off easily without changing code or software.

There are 79 data centers spread around the world in strategic points, so that on average a request takes less than 10 hops and 30 ms, resulting in a website with global presence. The nodes cache the static files to have them closer to the visitors request, while delivering the dynamic content directly from the web server. It also uses Anycast technology to route the visitors to their nearest data center. As a next-generation CDN, it makes your website compatible with protocols not globally used like IPv6 or HTTP/2 and SPDY, establishing a seamless connection.

CloudFlare network can withstand the loss of 50% of the network without impacting service availability. Serving 43 billion DNS queries per day, updates to DNS take less than a minute and changes in your website hosting or recovery configurations are near instantly updated. DNS can also resist large scale DDoS attacks by applying the proper protections based on experience, like rate limiting, filtering or blocking. It supports 2-factor authentication when logging in your DNS setting too.

CDN is vastly used by nearly every website on the Internet. In order to make it more efficient and optimized, CloudFlare deployed WCO (Web content optimization) services that rendered pages as fast as possible and improved the loading of all the resources. Combining this services with the CDN, performance increases and your website gets optimized both at the network and browser level.

Security in CloudFlare grows by learning from every attack suffered and sharing the information with the rest of the community. This collective knowledge

makes CloudFlare network smarter, regardless of the size of the website. SSL can be activated easily, without worrying about certificates and their maintenance, to add an additional layer of security to the connection with your visitors. For a basic use you are not required to change your existing configuration when adding SSL, but CloudFlare also offers more advanced configurations that support certificates from any CA (certificate authority), full end-to-end SSL with certificate checking, and Keyless SSL.

4

Cloud Simulators

The adoption and deployment of cloud computing is increasing and it has become a popular research field. Deploying real cloud infrastructures for testing is really expensive, but as it is critical to evaluate performance and security issues, an alternative has been developed through software simulators [5].

Cloud simulators provide this functionality. As a testing cloud-based software, the cost is minimal compared to hardware and it allows the testing of different scenarios, redoing the analysis until the desirable output is achieved.

Decreasing the complexity and separating quality concerns, simulators analyze system behavior by focusing on quality issues of specific components under different scenarios. As a result, a wide selection of cloud simulators is available so the user can choose whichever suits more to the desired analysis.

These are some of the cloud simulators available:

- **CloudSim**

  Build upon the core engine of grid simulator GridSim, and based on Java, CloudSim is an event driven simulator which has diverse features and allows many more through its easy extendability [6].

  It is able to model and create huge data centers, unlimited number of virtual machines, introduce brokering policy or support the cloud characteristic pay-as-you-go model. Furthermore, CloudSim shines over other simulators in its feature of federated policy, not available in most simulators.

- **CloudAnalyst**

  This simulator was derived from CloudSim, extending some capabilities, and focused in evaluating performance, and cost of large-scale Internet applications in a cloud environment having a huge user workload, based on different parameters.

CloudAnalyst has an attractive graphic interface (GUI) and provides flexibility to configure hardware parameters (memory, storage, bandwidth limit...) of a virtual machine or data center. Moreover, a modeler can repeat simulations and experiments variating parameters quickly and easily.

- **GreenCloud**

  Through the necessity of a simulator that analyzed the energy efficiency of the clouds, GreenCloud was created. It is an advanced packet-level simulator designed so it can calculate energy consumption by the data center IT equipment such as servers, links, switches...

  In addition to the packet level analysis, the simulator can provide the workload distribution in the cloud system. A GUI is not available, not taking into account the graphical results, it is necessary to know C++ and Otcal, and the simulation takes minutes to finish because is time and resource consuming.

- **EMUSIM**

  EMUSIM is different from other simulators in the fact that it provides both simulation and emulation of cloud applications.

  CPU-intensive applications that are very costly for actual deployment, require the analysis of the experiment before renting resources blindly. Simulating, the dependency is entirely on the characteristics of hardware and software, but with emulation the software model is tested in the actual hardware.

- **GroudSim**

  This platform is an event-based simulator, specially made for simulating scientific applications in both cloud and grid computing.

  It is mainly concentrated on the IaaS, has Java as the underlying programming language, and can be extended to support additional models like cloud storage or PaaS.

- **DCSim**

  DCSim concentrates on virtualized data centers deployed in IaaS, in order to evaluate and develop data center management techniques. The data center simulated have centralized management systems and their network topology is neglected for higher scalability.

  Contains multiple interconnected hosts with each having its own CPU scheduler and resource managing policy. Moreover, it supports virtual machine

migration and sharing of workloads between VMs running multi-tier applications.

Table 4.1 shows a comparison of the diverse features and characteristics of the mentioned simulators.

Simulator	Underlying platform	Programming Language	GUI	Communication Model	Support of TCP/IP	Cost Modeling	Simulation Time
CloudSim	SimJava	Java	No	Limited	None	Yes	Seconds
CloudAnalyst	CloudSim	Java	Yes	Limited	None	Yes	Seconds
GreenCloud	NS-2	C++, Otcel	Limited	Full	Full	No	Minutes
EMUSIM	CloudSim, AEF	Java	No	Limited	None	Yes	Seconds
GroudSim	-	Java	Limited	No	Full	No	Seconds
DCSim	-	Java	No	No	None	Yes	Minutes

Table 4.1: Comparison of simulators

4.1 Detailed energy consumption (Green Cloud)

Large-scale data centers, composed of thousands of computing nodes, have been deployed all over the world, consuming high amounts of electrical energy. Surprisingly, the supplied energy is not mostly used to provide power to the servers, but to maintain interconnection links, network equipment operations, and also to be used by air-conditioning systems so the infrastructure stays fully operative [7].

Green Cloud emerged as a solution for this problematic. This simulation environment studies the energy aware conditions of cloud computing data centers in realistic setups. It details a model of the energy consumed by the elements of the data center, such as servers, switches and links. Furthermore, there is a thorough investigation of workload distributions within the data center.

Following these lines there are two simulations done by Greencloud, based on the architecture represented in Figure 4.1, called Three-tier high-speed, composed in this particular case by 1 switch in the Core Layer, 2 on the Aggregation Layer, 3 on the Access Layer, and 144 servers with a Virtual Machine in each host.

In both simulations, the main difference is the use or not of the power saving technologies DVFS (Dynamic Voltage Frequency Scaling) and DNS (Dynamic Shutdown) [7, 8]. This technologies have the responsibility of reducing the supplied voltage on the servers that may be underloaded, in the case of DVFS, and put the idle servers into an sleep mode until they are required again, for DNS.

As can be seen in the results of the simulation in Figure 4.2, the use of both technologies to reduce the voltage provided or stop the server, causes a reduction of 15% in the energy consumed by the servers, making the whole system reduce its global consumption.

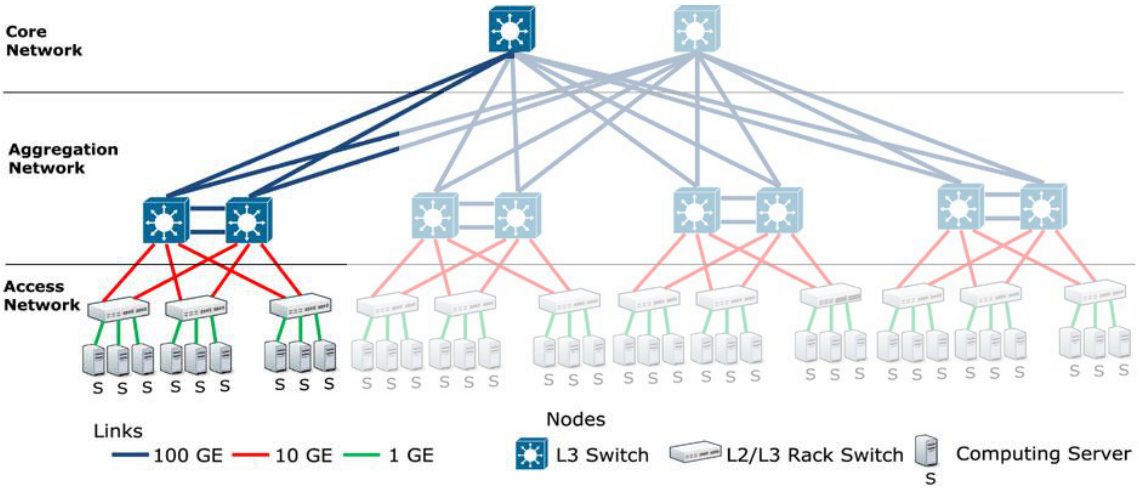

Figure 4.1: Three-tier high-speed data center

4.2 VM migration in Power Aware Data centers (CloudSim)

As previously noted, the management of data centers has changed towards awareness of power consumption. Idle servers can consume up to 70% of their peak power due to their narrow dynamic power range, so when servers are not used, the system is highly inefficient.

In order to solve that, virtualization and migration of virtual machines was adopted. Consolidation of VMs in a server allows other servers to power off, ending their consume of electricity. It is important that this operations are done without significant interruption of the service or performance degradation, because the service provider has to always comply to the stipulated in Quality of Service (QoS) defined in the Service Level Agreement (SLA) [9, 10].

There are usually three steps in any migration algorithm of virtual machines: VM detection, selection, and placement.

- **VM detection for migration**

  To decide when is the correct time to migrate VM from a host, different techniques are used based normally, on a static or dynamic threshold that stipulates if the host is overloaded or underloaded.

  - **Static Threshold (THR)**

    Static upper and lower thresholds are established and if the utilization of the host overcomes any of these limits, the system detects it. It is not the most suitable technique due to the unpredictability of the workload.

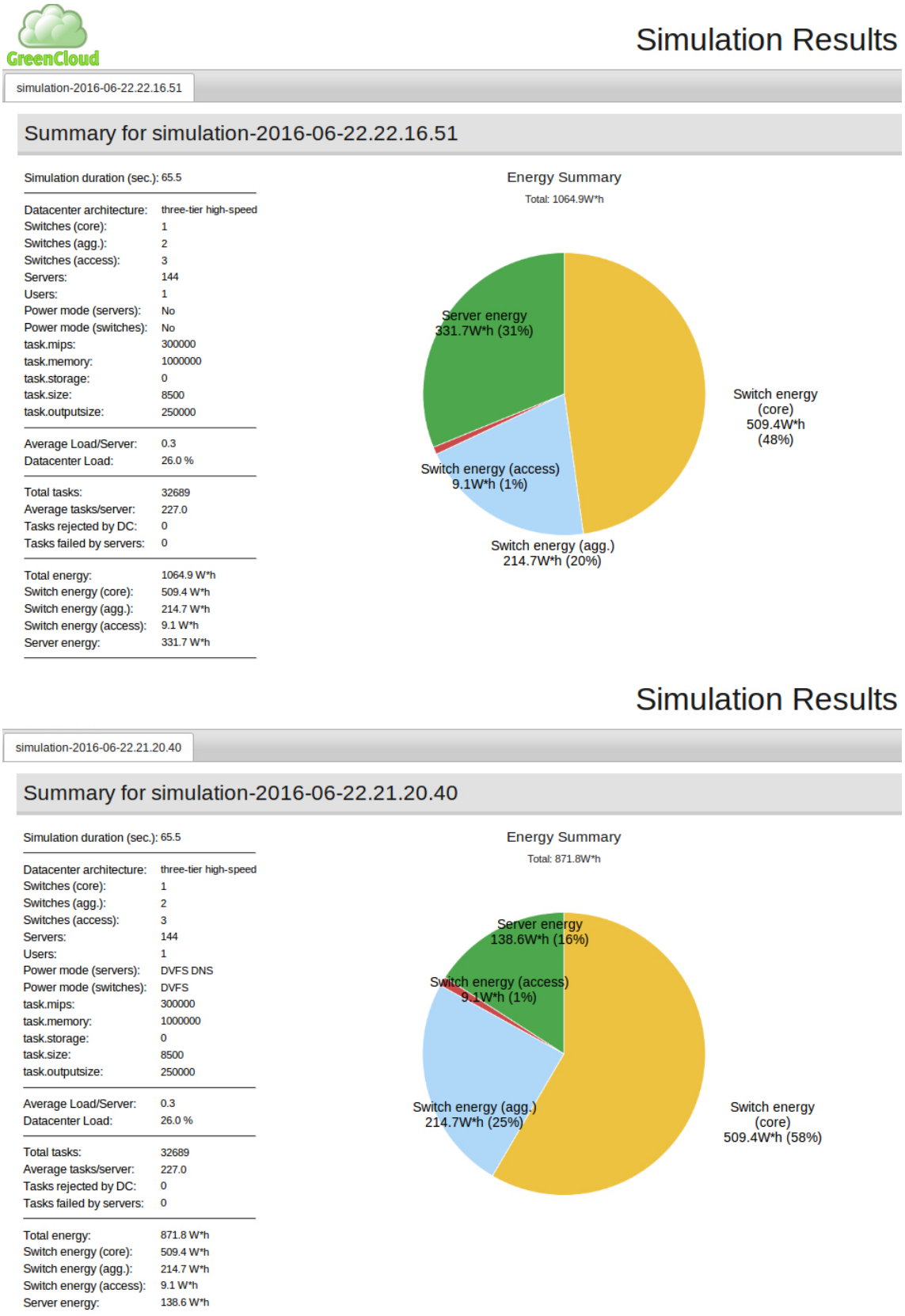

Figure 4.2: Simulations results for NonPowerAware (above) and DVFS+DNS (below)

  - **Median Absolute Deviation (MAD)**

    Statistical dispersion for adjusting the upper bound. Since MAD is not greatly influenced by the outliers (atypical value distant from other observations in an experiment), it is preferred to standard deviation.

  - **Interquartile Range (IQR)**

    Method similar to MAD to establish an adaptive upper threshold. In case that the distribution is symmetric, half of IQR is equal to MAD.

  - **Local Regression (LR)**

    Builds a curve that approximates original data by setting up the sample data models to localized subset of data.

  - **Robust Local Regression (LRR)**

    Iterative method created from Local Regression due to the vulnerability of the latter to outliers. Bi-square, a robust estimation method was added to the previously technique to provide a more consistent solution.

- **VM selection for migration**

  Having found the overloaded or underloaded host, the next step is to select, through different policies, the VMs that will be migrated from one host to the other.

  - **Maximum Correlation policy (MC)**

    In this policy, the correlation between VMs is found and the one with maximum correlation is selected for posterior migration.

  - **Minimum migration time (MMT)**

    By comparing the time it would take to migrate each VM allocated to the host, MMT selects the one that requires the minimum.

  - **Maximum Utilization (MU)**

    MU selects the VM that consumes maximum CPU capacity to migrate it to another machine.

  - **Random Selection Policy (RS)**

    A random number (uniformly distributed discrete random variable) is generated and according to this number a VM is selected for migration.

- **VM placement**

  There are different solutions for VM placement but the most popular, based on a bin packing problem, is the Best Fit Decreasing (BFD) algorithm.

CloudSim, the popular simulator, uses a modification of the previous algorithm called Power Aware Best Fit Decreasing (PABFD). This new algorithm sorts the VMs in a decreasing order, according to their CPU utilization in an specific time. Then for each VM, starting from the one with more CPU utilization, finds a host for which the increase of power consumption is minimum. The algorithm ends when all the VMs of the migration list are placed in other hosts.

Technique	Simulation time	Energy Consumption	Number of VM migrations	Overall SLA Violation	Number of host shutdowns
Non-Power Aware	86400s	150.68 kWh	0	0	29
DVFS	86400s	52.98 kWh	0	0	29
THR	86400s	41.81 kWh	4839	3.25%	1424
MAD	86400s	45.61 kWh	5265	1.31%	1528
IQR	86400s	47.85 kWh	5502	1.05%	1549
LR	86400s	35.37 kWh	2872	3.16%	806
LRR	86400s	35.37 kWh	2872	3.16%	806

Table 4.2: Simulation results when migrating VMs for different detection techniques with MMT selection policy

Policy	Simulation time	Energy Consumption	Number of VM migrations	Overall SLA Violation	Number of host shutdowns
Non-Power Aware	86400s	150.68 kWh	0	0	29
DVFS	86400s	52.98 kWh	0	0	29
MC	86400s	46.86 kWh	5085	1.13%	1517
MMT	86400s	47.85 kWh	5502	1.05%	1549
MU	86400s	49.32 kWh	5789	0.98%	1622
RS	86400s	46.84 kWh	4949	1.06%	1490

Table 4.3: Simulation results when migrating VMs for different selection policies with IQR detection technique

Table 4.2 illustrates the simulation with CloudSim of a cloud network with 50 hosts and VMs, in the case of: a Non-Power Aware datacenter, a datacenter only applying DVFS, and different cases depending the detection technique used, with MMT policy for all of them.

As can be seen in the results, the Energy Consumption is diverse, not only between non-power aware and energy efficient datacenters, but among the different techniques to detect overloaded hosts. Local Regression provides the most energetically efficient result, with the minimum migrations and shutdowns too, but the higher SLA violations have to be considered.

In the other hand, in Table 4.3 the simulation of the same system with different selection policies is provided (taking IQR as the VM detection policy), and the results in both energy consumption and VM migrations are quite similar, with RS as the one more efficient, due to its low energy consumption, and that it requires fewer VM migrations and host shutdowns for approximately the same percentage of SLA violations.

After analyzing the different simulators provided, and running a few simulations to verify their functionalities, it is safe to affirm that these tools are really useful for the planning of Cloud Computing networks, and their existence helps their users deploy and invest safely and securely.

It is necessary, though, to further develop and expand this measure tools, since they still have many limitations, and most of the time require extensions to upgrade their capabilities.

5

Security: challenges and solutions

Cloud computing offers improved, optimized and low cost service for costumers, using diverse technologies like virtualization or multi-tenancy. But apart from the security risks shared with conventional IT infrastructures, it adds cloud specific risks. The two previous technologies allow the use of the same pool of resources by multiple users. but they also introduce certain risks in the system.

This challenges and possible solutions, found in the research literature, are described in the following subsections [11, 12, 13].

5.1 Communication

Challenges:

- **Shared communication infrastructure**

  Attackers take advantage that resource pooling implies sharing network components, to execute cross-tenant attacks. Cloud Service Providers don't allow vulnerability scans or IP segregation of the network to avoid possible threats.

  Another form of attack is acquiring the system IP or MAC addresses and make malicious use of the network interfaces. Having the attacker super-user access, the real network ends up suffering from sniffing and spoofing.

- **Virtual network**

  Virtual networks are logical networks built over physical networks that are responsible for communication between VMs. Malicious activities of the VMs are not detected because security tools are only able to monitor the physical network, and IDS or other prevention mechanisms usually depend on traffic patterns that they now cannot detect.

As a result, the multiple VMs that share the virtualized network can receive attacks like DoS (Denial of Service), spoofing, sniffing, leakage of cryptographic keys, data breaches, etc.

- **Security misconfigurations**

  Misconfigurations are of significant importance in providing secure cloud services, and even the smallest one can result in a breach of security.

  The configurations have to be thoroughly managed to cover all the security requirements , and changes in the security policies of the cloud network should dynamically update the configurations to avoid session hijacking and loss of sensitive data.

Solutions:

- **Advanced Cloud Protection System (ACPS)**

  ACPS provides multiple security services to the CSP, like neutralizing cross tenant attacks through monitoring the VMs running at host platform, or providing auditability for the actions of VMs. It's prototype was implemented on Eucalyptus and OpenECP, both open source cloud platforms.

  The system is divided into multiple modules located at the host platform: the interceptor detects suspicious activities at the host; the warning recorder stores this recorded activities in the warning pool; the evaluator assesses the activities; and finally, the actuator reacts following the security policies if the amount of warning generations increases substantially.

  The periodic checksum verification also keeps the cloud entry points constantly monitored. The ACPS computes the checksums for critical infrastructure at setup time and compares them with the re-computed checksums, sending a warning to the evaluator in case of anomalies.

  It's also highly important its ability to remain transparent and undetectable to the VMs. Since the interceptor module does not block the initial system call, it does not get detected, and when the attack activity is confirmed, actions are taken.

- **CyberGuarder**

  CyberGuarder provides virtual network security through the deployment of virtual network devices, and transmits the data between VMs utilizing a layer-two tunnel VPN (Virtual Private Network). The data is transmitted without transiting through the central server but its metadata is stored to optimize traffic between VMMs.

Software ports are designed to monitor network traffic and security systems like IDS (Intrusion Detection System) are adaptively deployed for security of applications running on the virtual network. Moreover, it also provides VM security through the integrity verification of applications and by monitoring system calls.

- **Model for virtual network security**

  The proposed virtual network model used the Xen hypervisor utilizing both the bridge and route modes to prevent sniffing and spoofing. In bridge mode the VMM attaches the VM directly to the virtual Ethernet bridge and this virtual device connects to the physical network. In route mode a P2P link is created between the VM and the domain 0 (the VM management domain).

  The model is divided in three layers: routing, firewall and shared network layer. The routing layer establishes a dedicated channel between physical and virtual network and assigns a unique logical ID to each channel that helps monitor the source of packets originating from the shared network. The firewall layer safeguards against the spoofing attacks from the shared network and isolates all virtual interfaces connected to the it, so they cannot communicate with other virtual shared networks. Furthermore, it does not allow the packets to update the routing table, so such packets end up discarded. The shared network layer, finally, prohibits the communication between VMs belonging to different virtual network channels.

- **Tree-rule firewall**

  Conventional listed-ruled firewalls were proved to be quite insecure regarding shadowed rules, redundant rules and swapping positions. Furthermore, its sequential rule searching and arrangement of bigger rules after the smaller ones, decreased its performance.

  To improve this technique a tree-rule firewall was proposed, were the first attribute of the packet header gets compared with the root nodes of the tree and, after finding the match, continues to the next level checking for the next attribute. This process stops when the firewall reaches the specified security policy for the given attributes.

- **DCPortalsNg**

  DCPortalsNg is a SDN (Software-Defined Network) technique to isolate virtual networks for various VMs.

By interacting with the open stack through a neutron plugin, it obtains all the information of the virtual network and builds its own data by mapping networks and tenants and assigning a unique identifier to each of the VMs.

This technique rewrites the packet, through opening it and extracting the source and destination addresses, to obtain network isolation. Only if the packets are destined to the same network they will be processed, discarding the others. After a valid transmission, the OpenFlow message is sent to the appropriate virtual switch to rewrite the packet with source/destination IP addresses replaced with identifiers. Moreover the MAC addresses are replaced by the MAC addresses of the physical host.

Through DCPortalsNg cross tenant attacks on the virtual network are avoided, as well as cross VM denial of service (DoS) attacks.

- **SnortFlow**

  Utilizing the Snort and OpenFlow systems researchers developed SnortFlow, a system for intrusion prevention over cloud environment build and tested over Xen-based cloud. It showed great performance in terms of traffic analysis and prevention against intrusion.

  The snortFlow demon collects all suspicious traffic and warns the interpreter, that analyzes the alert and invokes the rule generator. This module then, develops the rules for the suspect traffic and forwards them to the OpenFlow device, which reconfigures the network according to the developed rules.

5.2 Virtualization

Virtualization allows the use of same physical resources instantiating a VM for each user providing them a complete operating machine. A VMM (VM monitor) or hypervisor manages the VMs, mapped to the same physical resources, and allows various operating systems to run simultaneously on the same physical system.

Challenges:

- **VM image sharing**

  As the method to instantiate VMs, images can be uploaded and downloaded from repositories in order to provide simplicity to the user. Malicious users can take advantage of this common practice and investigate the

code of the image to find an attack point. It can also be possible to recover some confidential information if the image was not properly cleaned.

Additionally, attackers can upload already infected images to introduce malware in the cloud computing system, when the image is instantiated, and by that monitor data and activity of other users.

- **VM isolation**

  Isolation is needed when instantiating VMs to the same hardware, being it storage devices, memory or computational hardware. Logical isolation is present but having access to the same physical resources could provide the malicious user with the opportunity of getting data or deploying cross-VM attacks.

- **VM migration**

  Load balancing, fault tolerance and maintenance can lead to migrating a VM to other physical hardware without shutting it down. This phase is crucial and needs to be done securely because the contents are exposed to the network and the code of the VM becomes vulnerable to attacks.

  This module can be compromised, provoking the relocation of the VM to a server or VMM under the power of the attacker.

- **Hypervisor issues**

  VMM or hypervisor is the key module responsible of the management and isolation of VMs, as well as generating and managing virtual resources. By attacking the VMM, the attacker can gain control of all the VMs managed by hypervisor, and extract all the metadata.

  Having multiple entry points and interconnection complexities, there has been reported bugs that let the attacker control the VMM in hypervisors like Xen, Microsoft Virtual PC and Microsoft Virtual Server, to gain privileged rights.

Solutions VM image sharing:

- **Mirage**

  As an management system for images in the cloud, Mirage regulates the publishing and retrieval of VM images through and access control framework provided at check-in and checkout times. Filters are applied to the

images at publishing and retrieval time to remove any leftover private information, malware, and pirated software. There is also a tracking mechanism used to keep track of an image for future auditability of actions and derivation

Furthermore, Mirage provides maintenance of the repository of images and executes periodic running of malware detection tools for the images and to discover vulnerabilities and patches.

- **EVDIC**

  Encrypted Virtual Disk Images in the Cloud (EVDIC) encrypts VM images on the disk, stores integrity information for the images, providing confidentiality and integrity services, and protects any sensitive data loaded into the image.

  The scheme first encrypts the image with the image encryption module when a VM is terminated using AES (Advanced Encryption Standard) with a key size of 256 bits, and then stores the encrypted image on the disk. The key is generated by a third party key management server through the password of the user.

  When retrieval is required, the image decrypt module interacts with the key management server to retrieve the decryption key and decrypts it for loading into a VM.

- **Scheme for patch management**

  This scheme checks for outdated software and vulnerabilities in the VM images in both possible states: live and dormant. To do that, the scheme is composed of two modules called the update checker and the OPS (Online Penetration Suite).

  The update checker keeps record of all of the software being used by VMs in the cloud setup, as well as version numbers and update releases, and it also is invoked periodically to scan the VMs. By matching the checked software with the installed and available packages, the scheme can detect obsolete software. The OPS examines the VMs for software vulnerabilities by using reputable security practices.

  Reports are made from the results of both modules to inform the user of the VM and the system administrator. The scheme provides up-to-date VMs to the user and avoids outdated software running in the administrators system. To update the VMs, it is necessary to do it manually.

- **ImageElves**

  ImageElves provides updated software installs and patches for the VMs both on the running and dormant.

  The technique keeps record of all the software running on the VMs, checks for updates, and identifies the VMs with need of an update, grouping them in similar classes. Then, it automatically updates the VMs by first installing it on a single VMs while making an image, and, if the update is successful, the image is applied to all the other VMs on the same class.

- **OPS-Offline**

  Offline Patching Scheme (OPS-Offline) identifies and rectifies dormant images in the image repository, with outdated software and malware vulnerabilities.

  The first module, the collector, downloads the images from the image repository and scans them to detect outdated software or presence of malware. The other module, the patcher, runs after the collector and patches the vulnerabilities.

Solutions VM isolation:

- **Secure runtime environment**

  The proposed architecture secures the VMs during execution time by untrusting management domain (Dom0) of Xen virtualization structure, protecting the user domain (DomU) and its private information.

  It denies memory access from Dom0 to DomU if not granted by the latter, and even in that case, these accesses are monitored by the hypervisor. The hypervisor also encrypts confidential memory regions to hide private information, and checks the integrity of the DomU, only allowing to restart if everything seems correct.

- **CloudVisor**

  This light weight security model provides privacy and integrity to the VM resources during runtime, by working under the hypervisor. From there, it intercepts the control transitions between the VMs and the hypervisor and performs security operations, like encrypting the general purpose registers or monitoring the address translation.

- **HyperCoffer**

  HyperCoffer trusts no component but the processor chip, with the memory data encrypted by the secure processor technology. The encryption is done with Address Independent Seed Encryption (AISE) while Merkle Tree is used for integrity checking.

  Like CloudVisor, each cache line is tagged with a unique VM identifier, preventing cross VM attacks, and it also intercepts control transitions. VMshim is the software that stands between VMs and the VMM that monitors and secures the control transitions and performs encryption/decryption.

- **CloudSec**

  This approach uses VM Introspection (VMI) to monitor the VMs physical memory externally. First it identifies the memory layout of the VMs hardware by analyzing the control registers of the VMs CPU. After that, through the hypervisor, it requests for Kernel Structure Definition (KSD) that maps to the physical memory bytes. Finally, the memory pages monitored have installed the memory access and time based triggers.

  The execution of the VM is halted if it receives a monitored memory page, and after loading the pages to KSD, the security of the new state is evaluated.

- **Exterior**

  Exterior is an architecture that launches a Secure Virtual Machine (SVM) that executes the kernel that is similar to the one of the Guest Virtual Machine (GVM). It redirects and updates the memory state at the hypervisor from SVM to GVM, resulting in the impression that the program is running in the modified GVM. Thanks to that, all the programs can be run externally of the OS.

  Additionally, the kernel data rootkit attacks and intrusions are detected by inspecting the code in the SVM, and removed from the GVM.

Solutions VM migration:

- **Secure migration mechanism**

  This migration technique is performed only if the destination platform is secured, taking into account a Trust Assurance Level (TAL) specified by the user when launching a VM.

  The TAL (least, low, average, normal, high) is computed using the credentials of the TPM, that measure the trust level of the hardware, and the Trust

Token credentials, that specify the trust level of the software stack. The migration is only allowed if the TAL is in the range specified. To trust the certifications Platform Trust Assurance Authority (PTAA) is assumed.

- **vTPM**

  The integrity and security conditions of the remote host are verified before migration. Next, the hosts establish a secure channel by mutual authentication and establishment of a session key. The VM that has to migrate gets bounded to a Virtual TPM (vTPM) that certifies its integrity before and during migration.

  By adding the vTPM, not only the integrity of the destination platform is ensured, but the migrating contents on the secure channel as well.

- **Framework for secure live migration**

  The proposed framework takes advantage of trusted computing to verify the attestation and integrity of the source and destination platforms. VM hopping and useless migrations are avoided thanks to the use of role based access control policies. Additionally, data during transmission is protected through encryption and digital signature.

  Regarding firewalls, the framework implements a per-VM firewall to control the communication, along with a host based firewall and IDS to provide network security.

- **Framework for security context and migration**

  This framework migrates the VM state and the static and dynamic security contexts to ensure the same security at the destination host.

  First, the static security context is sent, followed by the VM state, and ending with the dynamic security context. By following this scheme, the destination host acquires the same level of privacy and integrity.

Solutions Hypervisor issues:

- **DeHype**

  The proposed scheme divides the hypervisor into two components to secure it and other system components and resources. The need for this is because the hypervisor runs in privileged mode, putting in danger all the system in case of attack.

  The de-privileged component is executed in user mode, and after decoupling the code of the hypervisor, the smaller modules that don't interact

with the OS are sent to this component. On the other hand, the portions that interact with the OS are replaced by the user-mode equivalents, and if not possible to move, kept privileged in another module called HypeLet. This module reduces the risk of compromising the hypervisor since most of its code does not have privileges.

- **HyperLock**

  HyperLock provides an isolated runtime environment for the VMM, restricting it from obtaining direct access to the host system. If there is an access to the host system the HyperLock regulates it.

  Furthermore, each VM is paired with a separate shadow hypervisor so that in the case an hypervisor becomes compromised, the only affected VM will be the one paired.

- **SplitVisor**

  The hypervisor gets divided intro two submodules: Guestvisor and Splitvisor. The Guestvisor runs in non-root mode and emulates the hardware for the VMs. The Splitvisor, gets executed in root mode and isolates the possible multiple Guestvisors. In case there is functions not desired in the code of the hypervisor, the users can add or exclude this functions from the Guestvisor.

  This technology not only reduces the trusted computing base but also limits the capabilities of the hypervisor in root mode to safeguard the system.

- **NoHype**

  In NoHype multiple VMs can be run with the hypervisor eliminating the attack surface on it completely.

  The system pre-allocates resources so the hypervisor does not have to manage them. It also avoid emulating I/O devices by virtualizing them. At boot time, guest OS gets temporary modified as a hypervisor, to check for available system configurations and resources. Finally, indirections are avoided because of the hardware and dedicated cores for the guest VM.

5.3 Data/storage

Challenges:

Users don't have full control over data like in conventional computing model, just some level of control on the VMs. Management of data and servers resides in the service providers, resulting in greater security risks.

In an environment shared by multiple users, the data is much more vulnerable to risks in terms of confidentiality, integrity, and availability, and this security risks are enhanced with the increasing number of users and applications. If one entity is weak, an attack on it can result in the unauthorized access to the data of all the users in the system. Furthermore, employees of SaaS providers can access the information leading to a potential violation of integrity.

While data is being processed there is also the possibility of malicious attacks, due to the fact that virtualizing implies sharing physical resources among multiple tenants. On top of that, if the backup data is outsourced to a third party, the risks grow incredibly.

Finally, not standardizing secure key management techniques does not allow the standard cryptographic mechanisms to scale well to the cloud computing model leading to probable risks to the data.

Solutions:

- **SecCloud**

  SecCloud encrypts the user data uploaded into the cloud to secure its storage and the computations performed on it.

  Keys for the user, cloud, and a trusted third party are generated through bilinear pairing. Then, the data (signed by the trusted third party) along with the verifiable signatures is sent to the cloud, encrypted using the user and cloud keys through Bilinear Diffie-Hellman. The cloud decrypts the data, verifies the signature and stores the data at the designated partitions.

  The computational security is ensured when the verifying agency verifies the results by rebuilding the Merkle hash tree, using probabilistic sampling (instead of rebuilding the whole tree).

- **Scheme for security of resident data**

  There are three proposed partitions in the cloud: public, private and limited access. The user rates between values of one to ten, the requirement of confidentiality, availability, and integrity. The values are used to determine Sensitivity Rating (SR) of the user data, and based on this value, allotted in one partition. Below three, the data is stored in the public partition, from three to eight, in the private, and above eight, in the partition with limited access.

  An index is prepared and encrypted to employ searching capabilities over encrypted data. It is sent with the encrypted data (128-bit SSL encryption with MAC appended afterwards) to the cloud and stored depending on the

SR value. To download the data from private and limited partitions, and not for the public partition, it is necessary user authentication carried out by the data owner and the cloud.

- **Methodology for security of resident data**

  The proposed methodology conducts the verification of the cloud data correctness without explicit knowledge of the whole data. Homomorphic tokens are pre-computed by the user and the data is fragmented and stored redundantly across the cloud servers. To verify data correctness, the cloud compares received random data blocks indices with the pre-computed tokens, and responds with a decision.

  Furthermore, the scheme performs error localization by detecting the misbehaving server, and insertion, deletion, modification and appending of data blocks is also supported. Security against cloud storage threats like integrity attacks, Byzantine failures and server colluding attacks is provided by this methodology.

- **FADE**

  FADE uses both symmetric and asymmetric encryption. The symmetric keys are protected using Shamirs (k,n) scheme.

  The protocol works with a group of KM (Key Managers) that act as a trusted third party. The data key (K) encrypts the file F of the client, and its encrypted as well by the symmetric key S. The group of key managers generate a public/private key pair (e,d) to encrypt S. A policy P postulates the policies under which access to the file is valid. In order to upload the data, a user requests the KM to generate a key pair by sending P. The KM generates public/private key pair associated with the P and transmits public part to the user. The user encrypts the file with randomly generated K and encrypts K with S that is further encrypted with the public key generated by the KM. At the end the whole encrypted package is stored at the cloud along with the P.

  When decrypting, all the data is downloaded from the cloud and S is sent to the KM for decryption through blinded RSA. As a result other keys and subsequently F is decrypted. Upon the expiration of the policy, FADE ensures policy renewal and revocation, with the KM deleting the corresponding keys and P, through secure overwriting that makes the data inaccessible and therefor assuredly deleted.

- **TimePRE**

  To share data securely in a group, forwarding it to the group users, and dealing with user revocation, TimePre was proposed. It consists of a time based proxy re-encryption combined with Attribute Based Encryption (ABE) that, without necessity of the data owner being online, revokes users and generates new re-encryption keys. This is done by associating the time period with every user, and upon expiration of the time period the user is automatically revoked by the CSP. The re-encryption keys are generated by the CSP through a pre-shared master key between the data owner and the CSP.

  As for the access control, ABE ensures the identification of users by set of attributes rather than identity. It uses eligible time periods for a user along with other attributes to identify a user. As a result of using this scheme, privacy and availability of the data within a group is ensured, but data integrity escaped from the focus of the proposal.

5.4 Cloud applications and API

Challenges:

Cloud applications must be used and managed over the web, so that without being bonded to specific users, they can always access them ubiquitously.

Despite inheriting the same vulnerabilities as traditional web applications, the ones in the cloud are far more insecure and vulnerable than the traditional, and its security solutions are not adequate to apply. Some of the risks identified are [14, 15]: Injection (SQL, OS, LDAP); Broken Authentication and Session Management; Cross-Site Scripting (XSS); Insecure Direct Object References, etc.

APIs bridge the users and the services in cloud computing, like an user guide that describes the details about the CSPs cloud architecture and features. The CSPs publish their APIs to show the features of their cloud and to market them, and in addition it allows users to build or extend the services using it.

On one hand, publishing APIs helps users know about the functionality of the cloud, but in the other the architecture is exposed to attackers, who may exploit vulnerabilities like weak credentials, insufficient authorization and input-data validation.

Solutions:

- **Diameter-AAA**

  This protocol protects the cloud applications by filtering the unauthorized access by employing network based access control.

  The requests are received by the network access server, then forwarded to the diameter server, and checked for the authentication and authorization parameters. Based on the results, the request is granted or denied access to the application. Diameter-AAA also provides accounting services within the cloud.

- **TPM-ECC scheme**

  TPM and Elliptic Curve Cryptography (ECC) provides a secure platform for application execution in the cloud and recommends the use of encryption while moving applications between platforms.

  ECC generates the keys and stores them in the TPM configuration registers. The integrity of the platform is ensured before moving any application to it, as well as the integrity of the application, that is checked at the destination platform before launching.

- **SECaaS**

  Security as a Service in the cloud environment recommends the security services provided by different clouds and a manager cloud (independent cloud) that tracks these services.

  The user specifies the security requirements to the manager cloud that identifies the cloud(s) providing those services. The user application is then registered with these clouds providing security services. The protocol works at all levels (SaaS, PaaS, IaaS) and secures the services.

- **API management platform**

  The API management platform proposed, providing access control architecture for the cloud APIs, is based on the Open Authorization (OAuth), which is a access control mechanism based on tokens. It uses tokens instead of user credentials in order to access the resources, taking place for example when the applications use a token on behalf of the user.

  Through the APO management platform, the API provider registers and publishes the API and obtains a key for validating the tokens. When an API consumer wants access, there is a request for a token in the platform and when validated, a key is obtained. The consumer then, calls the API

by using the token signed with its private key and the provider sends the token to the platform for validation, granting access if valid.

5.5 Identity management and access control

Challenges:

Cloud computing deals with the fact that the owner and the resources are in different administrative domains, and organization's authentication and authorization may not be exported to the cloud. Both characteristics are important and cannot be satisfied by conventional identity management and access control systems.

There is need then, for a dynamic, fine-grained, and strict access control mechanisms to control unauthorized operations, and the control of organizations over identity management system, to update the access control policies in cases of change in the personnel.

Some of the issues caused by insecure identity management and access control are: denial of service by account lock-out; weak credential reset mechanism; insufficient authorization checks; cross domain authentication, insufficient logging and monitoring activities, etc.

Solutions:

- **HASBE**

  Hierarchical Attribute-Set-Based Encryption (HASBE) is an extension of Attribute Set Based Encryption (ASBE), which at the same time is an extension of Attribute Based Encryption (ABE). The latter is employed when providing access control in the cloud, specifying and enforcing the access control policies cryptographically. Basically, it associates the encrypted messages using the attributes and the user in possession of this attributes is able to decrypt them. ASBE, extends this functionality by categorizing the user attributes into a recursive set based arrangement and allows users to enforce dynamic constraints on how those attributes mutually fulfill access control policy. HASBE, finally, adds hierarchical user structure to the protocol.

- **Descentralized access control for cloud storage**

  This decentralized approach uses ABE for access control, and Attribute Based Signature (ABS) when anonymous authentication is desired. To authenticate a user without revealing its identity, the signature is computed and verified based on the attributes, avoiding authentication by identity.

Tokens are also issued by a third party to the users, that present them to the Key Distribution Center (KDC), whose in charge of issuing the keys for encryption/decryption and signing. With this keys, based on bilinear pairing, the user signs the encrypted data and transmits it to the cloud. The cloud verifies the attribute based signature and stores the data if the user is valid. If the data has attributes similar to that of a revoked user, the encryption parameters are changed and the user access is revoked.

- **RB-MTAC**

  Role Based Multi-tenancy Access Control (RB-MTAC) combines identity management and role based access control.

  Users have to register with the cloud to obtain a unique ID and set a password. To access the cloud the user needs to get identified through the identity management module, and if positive, it gets redirected to the role assignment module connected to the RB-MTAC database which assigns roles to the user based on registered role information. The RB-MTAC module maintains the access control list for resources and permits or denies access to them to the user.

- **SPICE**

  Simple Privacy-preserving Identity-Management for Cloud Environment (SPICE) provides anonymous authentication, delegatable authentication, unlinkability, accountability, and user centric access control through group signature and randomization.

  The user registers with a trusted party called the registrar and obtains a single credential for all the services provided by the CSP. Then, generates with the credentials an authentication certificate, being able to create different versions from the same credential, if desired. The group signatures are used over the certificates for authentication by ensuring that the signature is from a valid user of the group.

  Unlinkability is provided by applying randomization to the signatures, as well as hiding of unrequired attributes.

- **Identity management framework**

  Based on User Managed Access (UMA) protocol, this identity management framework has the infrastructure as the Authorization Manager (AM), the CSP as a host, and the services owner as an authorizing user.

  The AM controls the services, manages the requesting users' identities and grants or denies resource to any request to the services, taking into account

the access control policies. The proposed framework can manage the identity management and access control across multiple CSPs where the AMs coordinate with each other to provide these services.

5.6 Contractual and legal aspects

Challenges:

SLA (Service Level Agreement) documents specify the terms and conditions between the user and CSP. Issues like performance assurance, regulatory laws compliance, geographic jurisdictions, monitoring of contract enforcement, as well as security requirements are agreed upon in the SLA by the user. In case of ambiguities, it is harder to claim the loss at the CSP.

Legal issues also arise due to the presence of CSP resources in geographically different legal jurisdictions. Sometimes the data may be present in more than one location having different laws about digital security. Additionally, the issue about E-discovery, seizing the hardware of the CSP for investigations regarding a particular client, supposes another security threat that could lead on privacy breach of other users.

Solutions:

- **SecAgreement**

  Ws-agreement (web services agreement) defines the syntax and semantics of publicizing the competencies of the service providers, creates the template based agreements, and monitors the agreement acquiescence. SecAgreements is a framework that extends ws-agreement, to incorporate security constraints and metrics into the terms of SLA. This extended template articulates the security parameters and services for provision in the SLA and integrates the elements that quantify the risks of using specific cloud services.

- **Framework for reacting to change in security environment at runtime**

  In case of violation of security SLA, or cancelation of any of the security services, a framework to reduce security risks was proposed.

  Its functionality is based on an algorithm that performs risk-aware renegotiation. It renegotiates and scrutinizes the obtainable services at runtime as a replacement to the canceled or problematic service. Then updates the risk evaluation according to the changes in the SLA. Furthermore, it is capable of

negotiating cloud federations to lower the risk and achieve the customer's security needs.

- **SPECS**

  SPECS, an architecture that provides SLA-based security as a service is divided and focused on three stages of the SLA life cycle: negotiation, enforcement and monitoring.

  This approach attaches security parameters with the SLA to let the end user judge the security offerings and requirements and to oblige the CSP to provide explicit security. The SPECS articulates the architecture only, and makes use of established work to carry out the phases of the SLA life cycle.

- **Embedding security controls in cloud SLA**

  The proposal built a compliance vocabulary composed of SLA security terms, selected from various standard documents (NIST, the common criteria, the CSA), and the associated security controls that fulfill the corresponding security requirements. The vocabulary, represented as an XML schema, lets the organizations compare the security services of different CSP easier.

  Moreover, an ontology was also built to automate the process of negotiation and selection of better security parameters for the SLA. Ontologies used the concept of service matchmaking to differentiate between different offerings.

Taking into account all the challenges and solutions described and investigated in research, security in Cloud Computing has its flaws but there is a lot of proposals to improve it. From communication to contractual problems, Cloud Service Providers and clients have to work together to enhance the technology and its security. As seen in this chapter, most of the challenges are responsibility of the CSP, since they are the ones that have full control of the infrastructure and can escalate it to protect its content, but it is necessary that the client performs all the operations in a secure and responsible manner, and does not create vulnerabilities in the system due to reckless behavior.

6

Deploying a secure app on AWS

Hosting a web app at AWS is mostly the same as doing it in your own servers, with the differences that there's no physical access to the infrastructure and the need to use various services of AWS for the application to be functional.

The majority of web apps are composed of three tiers: the user interface via the web browser; the app functionality with the web server; and finally, the storage of data through a database server. As mentioned, and taking into account that the first tier is chosen by the client in their own infrastructure, the second and third are usually deployed on an EC2 and RDS instances (with the addition of services like Amazon EBS, Amazon S3, Auto Scaling, Elastic Load Balancing, Amazon CloudWatch, Amazon Route 53, and Amazon CloudFront to escalate and improve the network) [16].

An example of this architecture can be seen in Figure 6.1, with EC2 and Database instances deployed in a VPC (Virtual Private Cloud) within different Availability Zones to reduce failure, and its connections restricted by Security Groups. To build this architecture is necessary to follow some steps in order to deploy it in the more secure way, avoiding then attacks and malicious actions.

6.1 Security basics: IAM Groups/Users, Key Pairs and VPC

Avoiding insecurities in the access and connections between the different elements of the architecture is what separates the robust applications from the ones receiving attacks.

For that reason is basic to start creating users and groups to classify who has access to the services offered by AWS and which permissions each group possess. IAM (Identity and Access Management) service offers the possibility of doing

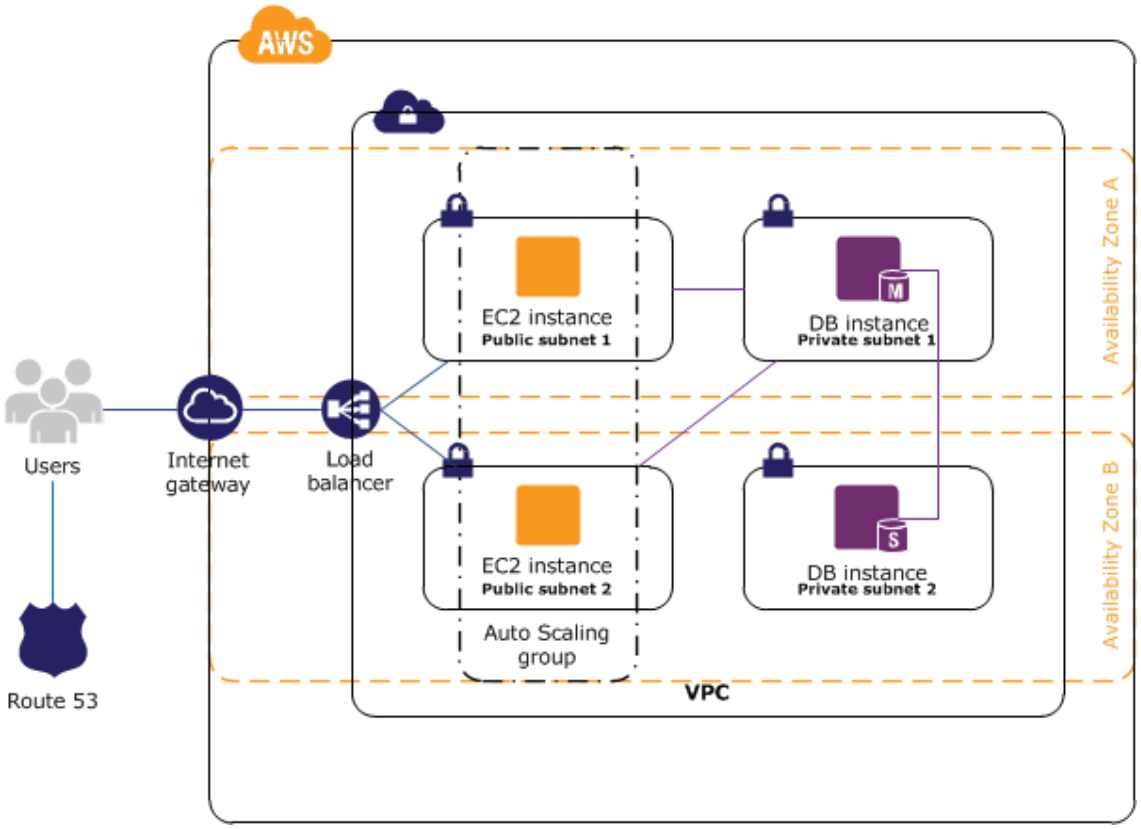

Figure 6.1: Example architecture for a webapp on AWS

both, allowing the administrators to limit the access to each service to the users appointed by them.

Additionally, Linux instances don't have passwords to log in, so AWS uses public-key cryptography to secure the login information through the association of the key pair with the instance when launching.

A key pair is a set of public and private keys that allow the encryption of data, such as a password, and the posterior decryption of data by the recipient. This method of authentication provides extra security that a simple password can not offer. The key pair, 2048-bit SSH-2 RSA keys, is associated with the instance at the time of creation, so only with it the access is possible. Furthermore, if multiple access to the instance is needed, and it is not desired that the users have root privileges, the creation of different key files allows solving the problematic.

Via the EC2 service you can create this key pair and download it in a secure location so you can later connect with SSH to your instance providing this key file.

Last, the creation of a VPC (Virtual Private Cloud). This virtual network simulates the environment you would have in a traditional network in your own datacenter, but with the capacity of escalating with the AWS infrastructure. It is

a logically isolated area within the AWS cloud where you can launch resources, like instances, in a secure manner, by using multiple layers of security. Configuration includes selecting the IP range, creation of subnets, and configuration of route tables, network gateways and security.

You can add 1 or more public subnets depending if you want multiple zone availability for the web servers, and then do the same for the databases but with the precaution of making them private, so only specific users can access them.

It is good practice to create the VPC with more than one availability zone defined. Failure in the AWS infrastructure is not common but sometimes possible within the parameters established in the SLA (Service Level Agreement), and for that reason all the precautions taken can provide a better result if there are problems.

6.2 Application Server: Security Group, IAM Role and EC2

Launching an EC2 instance allows to create a virtual server where you can run the webapp.

To secure the instances is necessary to assign Security Groups to them, that will act as a virtual firewall controlling which type of traffic is allowed or prohibited. Security Groups are based on rules that are created, updated and deleted any moment, taking effect immediately. When launching and instance, one or more security groups are associated with it, and if there is traffic that wants to reach the instance, all the rules from all the security groups associated with that particular instance are evaluated. This security feature provides a strong defense against attacks from outside the VPC, and in the case that the Security Groups wouldn't met the necessary requirements, the administrator can always add its own firewall to protect the instances.

In the case of a webapp there would be mainly two rules: one to allow inbound HTTP from anywhere, and the second to connect with SSH from the IP of the administrator.

In order for the applications deployed in the instances to securely make API requests to AWS, there should be different roles that specify what API actions and resources the application can access, and each instance should have a role assigned.

Finally, the instance can be launched when selecting the wanted AMI (Amazon Machine Image), Security Group, Role and key pair, previously created.

6.3 Database Server: Security Group, RDS

Similarly to the EC2 instance, the Database Server will have a new Security Group that will allow only access from the web server, making it unreachable from outside the infrastructure. To do that there should be a rule within the SG that allows inbound MySQL queries from the web server SG , that will include all the instances that have this SG assigned.

After that, with the Amazon RDS service there is the possibility to choose between various types of engines such as Amazon Aurora, MySQL or MariaDB, to launch the database with the respective Security Group assigned.

6.4 Deployment of the app: connecting and configuring

The deployment of the webapp, as in traditional networks, consists on connecting to the instance via SSH, with the particularity of the use of a key pair archive (.ppk) to log in. After that, the next step is to transfer the app files to the path "/var/www/html/" along ownership and configuration commands. Finally through the configuration of the app the administrator will be able to connect the application with the database stored in the Amazon RDS servers.

Throughout the installation, the user is able to create a custom Amazon Machine Image to later deploy in new instances in order to avoid repeating the same steps.

An example of the deployment of a website containing an application can be seen in Figures 6.2 and 6.3.

6.5 Enhancement of the architecture: Auto Scaling, Load Balancing and Domain Name

After configuring the network and all its elements, AWS offers various services to give the administrator the ability to enhance the capacities of the architecture.

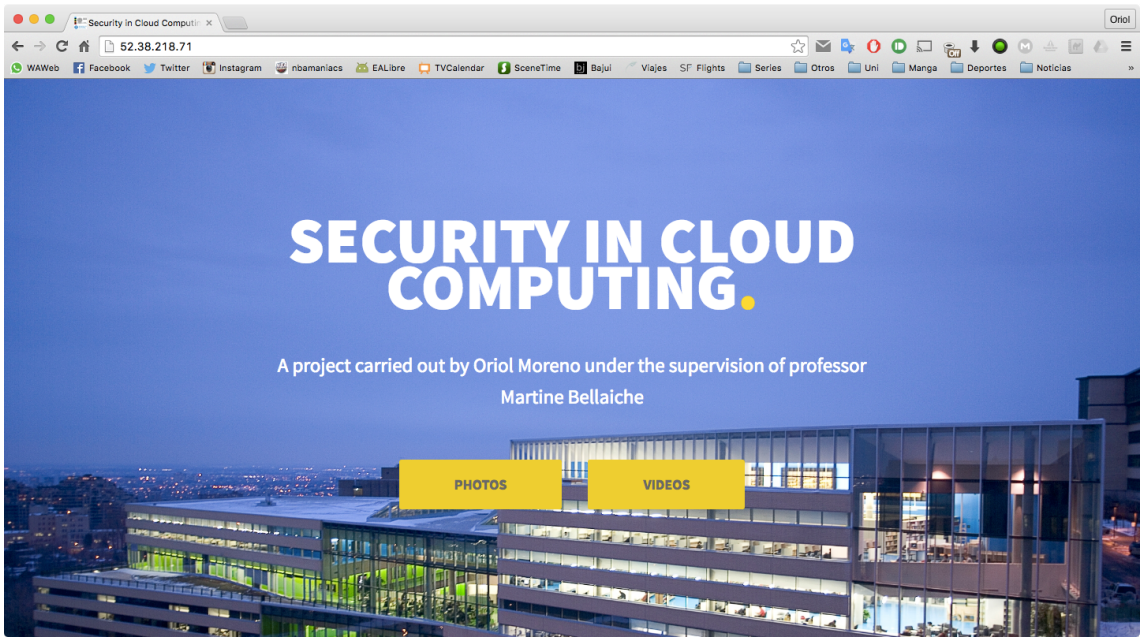

Figure 6.2: Wordpress website on instance 52.38.218.71

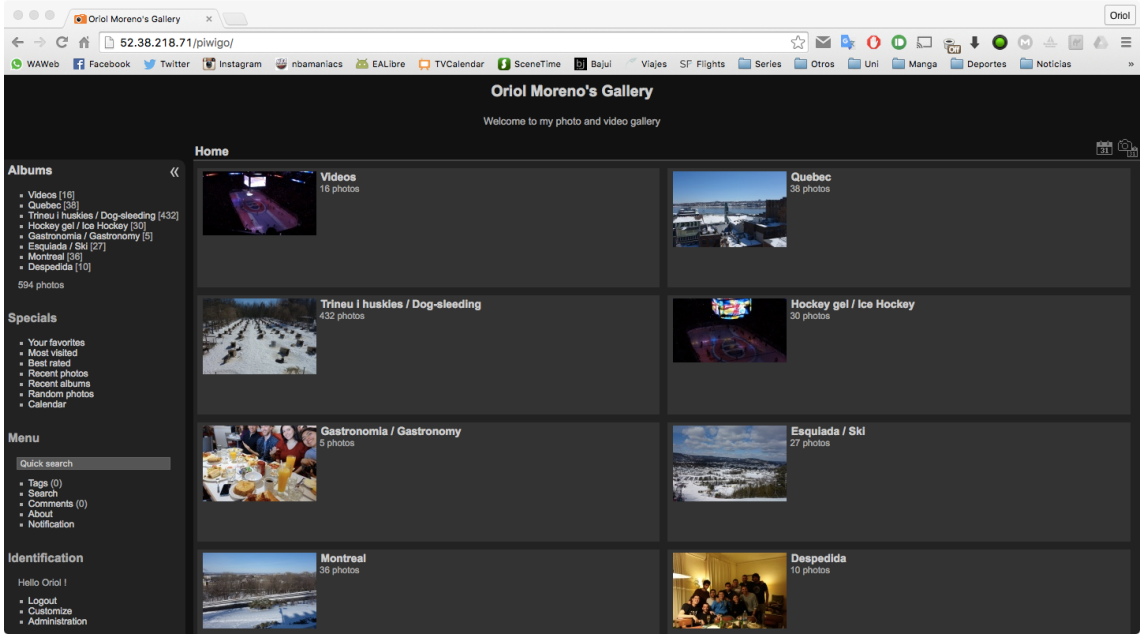

Figure 6.3: Piwigo webapp on instance 52.38.218.71

Auto Scaling allows to create policies to launch or terminate instances based on the needs of your application. Policies examples would be launching instances whenever the CPU usage surpasses an established limit or terminating instances over the weekend when the traffic is lower.

Additionally, Elastic Load Balancing distributes the income traffic through the designated instances for balance and availability purposes. As the applications requirements change the administrator can add or remove instances to the load balancer.

Finally, with Amazon Route 53, the instance Elastic IP can be associated with a memorable domain name to route visitors to the application.

7
Conclusions

Cloud Computing is the technology of present and future, that much is obvious. As the sharing and consuming of data all around the globe grows, having access to flexible and on-demand resources becomes more and more necessary. But the technology is far from perfection.

Business still see "the cloud" as a security risk. While small companies embrace the benefits cloud computing provides, mainly because of their lack of capital, big enterprises are reticent to move their private data outside their infrastructure, making its standardization slower.

The particular security risks of the technology, besides the ones inherited from traditional networks, have to be studied and solved as much as possible, making them one of the top priorities for researchers to investigate. Since its inception, the studies related to cloud computing have grown exponentially, resulting in being one of the majors fields of investigation in IT. And in the next years this situation will only get magnified.

Finally, it is also important to enhance the energy saving capabilities of the cloud networks. As the studies have shown, a datacenter consumes huge amounts of energy due to its inefficiency to save power in its components and links. If the technology is deployed all over the world, with great amounts of datacenters build, improving the current consumption is an obligation in order to waste less power and become more energy aware.

Bibliography

[1] Q. Zhang, L. Cheng, and R. Boutaba. Cloud computing: state-of-theart and research challenges. *J. Internet Services and Applications*, 1(1), 2010.

[2] Amazon Web Services: Overview of Security Processes Consulted at https://d0.awsstatic.com/whitepapers/aws-security-whitepaper.pdf.

[3] White paper: Microsoft Azure Network Security Consulted at http://download.microsoft.com/download/C/A/3/CA3FC5C0-ECE0-4F87-BF4B-D74064A00846/AzureNetworkSecurity_v3_Feb2015.pdf.

[4] CloudFlare - The web performance and security company Consulted at https://www.cloudflare.com.

[5] A Survey of Cloud Computing Simulations and Cloud Testing Consulted at http://students.cec.wustl.edu/~azinoujani/.

[6] A. Ahmed, A.S. Sabyasachi. Cloud computing simulators: a detailed survey and future direction. *IEEE International Advance Computing Conference (IACC)*, 2014, IEEE, 2014, pp. 866-872.

[7] Dzmitry Kliazovich, Pascal Bouvry, Samee Ullah Khan. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. *J Supercomput*, 2010.

[8] Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Calheiros, Rajkumar Buyya, Mihai Alexandru. Energy-aware simulation with DVFS. *Simulat. Model. Pract. Theory*, 39 (December) (2013) 76-91 (S.I. Energy efficiency in grids and clouds).

[9] Chowdhury, Mohammed Rashid and Mahmud, Mohammad Raihan and Rahman, Rashedur M. Implementation and Performance Analysis of Various VM Placement Strategies in CloudSim. *J. Cloud Comput.*, December 2015 4(1) 45:1–45:21.

[10] Madhu B. R, Dr A.S. Manjunatha, Prakash Chandra, Chidananda Murthy P.. A Comparative Study of Algorithms For Efficient Dynamic Consolidation of Virtual Machines In Cloud. *International Journal of Applied Engineering Research*, 11(6) (2016) pp 4597-4600.

[11] Kuyoro S. O, Ibikunle F, Awodele O. Cloud Computing Security Issues and Challenges. *International Journal of Computer Networks (IJCN)*, 3(5), 2011.

[12] Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina, Eduardo B Fernandez. An analysis of security issues for cloud computing. *Journal of Internet Services and Applications*, 2013, 4:5.

[13] Mazhar Ali, Samee U. Khan, Athanasios V. Vasilakos. Security in cloud computing: Opportunities and challenges. *Information Sciences*, 305 (2015) 357-383.

[14] Security guidance for critical areas of focus in cloud computing v3.0 Consulted at https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf.

[15] The Treacherous Twelve: Cloud Computing Top Threats in 2016 Consulted at https://cloudsecurityalliance.org/group/top-threats/.

[16] Amazon Web Services (AWS) - Cloud Computing Services Consulted at https://aws.amazon.com/.