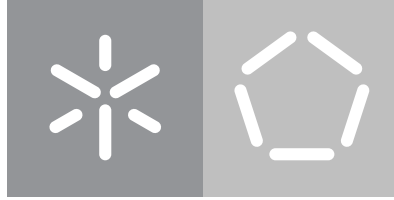


Universidade do Minho
Escola de Engenharia

João Aloísio Sousa Araújo

Icarus - A Cloud Security Perspective



Universidade do Minho

Escola de Engenharia

João Aloísio Sousa Araújo

Icarus - A Cloud Security Perspective

Master's Dissertation

Integrated Master's in Informatics Engineering

Work supervised by

José Carlos Bacelar Ferreira Junqueira de Almeida

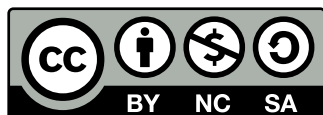
COPYRIGHT AND TERMS OF USE OF THIS WORK BY A THIRD PARTY

This is academic work that can be used by third parties as long as internationally accepted rules and good practices regarding copyright and related rights are respected.

Accordingly, this work may be used under the license provided below.

If the user needs permission to make use of the work under conditions not provided for in the indicated licensing, they should contact the author through the RepositóriUM of Universidade do Minho.

License granted to the users of this work



**Creative Commons Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional
CC BY-NC-SA 4.0**

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.pt>

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

Lorem ipsum.

Acknowledgements

I want to thank Celfocus for this opportunity and all the colleagues who helped me during the internship. Special acknowledgment to Professor José Almeida for his support and availability in supervising this study. And finally, thank my family and girlfriend for all their support.

Abstract

Increasingly, cloud computing is used because of its significant advantages. However, this use can increase risk, as the solutions are not in the organizations' infrastructure but in an external perimeter.

This thesis presents a study of cloud security in which an agnostic reference architecture is developed for any cloud service provider. The three most used providers are also compared in order to materialize the architecture and make a proof of concept.

The solution presented was based on the controls in Annex A of ISO 27001 (information security) and aimed to minimize the increased risk of applications hosted in the cloud as much as possible and speed up the process of any need to obtain ISO 27001 certification.

Keywords: Cloud, Security, Architecture, Controls, ISO ...

Resumo

Cada vez mais, a computação em nuvem é utilizada devido às suas grandes vantagens. No entanto, esta utilização pode vir com um risco acrescido, pois as soluções não estão nas infraestruturas das organizações mas, sim num perímetro externo.

Esta tese apresenta um estudo de segurança na nuvem em que é desenvolvida uma arquitectura de referencia agnóstica a qualquer prestador de computação em nuvem. São comparados também os três prestadores mais utilizados a fim de materializar a arquitectura e fazer uma prova de conceito.

A solução apresentada foi baseada nos controlos do anexo A do ISO 27001 (segurança da informação) e tem como objetivo minimizar ao máximo o risco acrescido das aplicações hospedadas na nuvem e acelerar o processo de eventual necessidade de obter a certificação do ISO 27001.

Palavras-chave: Nuvem, Segurança, Arquitectura, Controlos, ISO ...

Contents

1	INTRODUCTION	1
1.1	Context	1
1.2	Motivation	2
1.3	Contributions	2
2	STATE OF THE ART	3
2.1	CyberSecurity OverView	3
2.1.1	What is CyberSecurity	3
2.1.2	The importance of Cyber Security	4
2.1.3	Security by Design	5
2.1.4	Zero trust and Least privilege access	5
2.1.5	Defense in Depth	5
2.2	Application Security	6
2.2.1	The OWASP Testing Project	6
2.2.2	Testing Techniques	6
2.2.3	OWASP Top 10	9
2.3	Container Security	10
2.3.1	Virtualization	10
2.3.2	Major risk of containers technologies	11
2.4	DevSecOps	12
2.5	Cloud Shared Responsibility Model	12
2.6	ISO	14
2.6.1	ISO 27001-Information Security Management	14
3	REFERENCE ARCHITECTURE	19
3.1	On-Premise Reference Architecture	19
3.1.1	Controls applied	20
3.2	Cloud Service Provider agnostic architectures	21
3.2.1	Single VPC architecture	21
3.2.2	Multi VPC architecture	24

3.3	Cloud Service Providers	27
4	PROOF OF CONCEPT	30
4.1	AWS POC Architecture	30
4.1.1	Implementation	35
4.2	Controls in Action	39
5	CONCLUSION AND FUTURE WORK	43
	Bibliography	44
	List of Figures	46

INTRODUCTION

1.1 Context

Over the past two decades, technology has developed very quickly, shown to have critical, but also very positive effects on people's lives. This implication of people with and in technology allows entities and organizations to collect an enormous amount of data. This data represent quantities, characters or/and symbols that can make us put up and understand information about a particular group of people and its trends. This general and common interest and liability of people with technology benefits make us associate the 21st century as the Data era, data that is far from companies' physical perimeter and increasingly resides in the cloud.

The term cloud or cloud computing is very known, you don't have to be an expert or someone in the field to understand what cloud is used for and the potential that it has. Companies are increasingly predisposed to deploy solutions in the Cloud to use all its advantages, such as low-cost hardware maintenance and scalability. They are already switching their business and services to the cloud, and the covid-19 pandemic is accelerating the process. However, we must be careful because speed can be the enemy of perfection most of the time. By 2024, the predictions are that 45% of IT spending on system infrastructure, software infrastructure, application and business process outsourcing will shift conventional solutions to cloud solutions (Pettey, 2020). However the famous saying "The cloud is just someone else's computer" has some basis and the very easy also entails some additional care and risks.

Moving services or products to the cloud come with some concerns in terms of security, with their data stored and processed in the cloud they are more exposed once the attack surface is higher the risk of unwanted disclosure or attacks increases as well. With a higher attack surface, new kinds of attacks are coming and the companies need to be prepared to that if they want to have their services in the cloud be safe and minimize the impact of a potential attack. They have to consider that a successful attack can have a huge impact, put their reputation at risk, and have financial and regulation costs.

1.2 Motivation

Celfocus is a Technology company that accelerates Product & Service innovation by promoting innovative digital capabilities and delivering business value in the most complex, mission-critical challenges. The main fields are telecommunications and e-commerce.

Celfocus always have security as a huge concern. They try to deliver their solutions as safe and secure as possible. A specialized security team and department search for new types of attacks, protection strategies, vulnerabilities, security methodologies, etc.

In the past years, the company cloud solutions have increased. They've needed to understand better how security is applied in a cloud model to ensure their application doesn't lack security when implemented in the cloud. This project, "Icarus- A cloud Security perspective," comes with that. The goal is to study how cyber security is handled nowadays and follow the various entities that assure information security, like the International Organization for Standardization or the National Institute of Standards and Technology. With that knowledge, apply the best practices, security methodologies, processes, and strategies to develop a safe and secure cloud infrastructure to host their application and solutions.

1.3 Contributions

This thesis presents a cloud reference architecture development, where the main concern is security, to host any applications or solution. The main goal is to migrate standardized security controls and best information security practices to the cloud to get a secure infrastructure. With that, ensure that a cloud solution doesn't need to be less secure than one implemented on a traditional data center. The specific contributions are:

- Develop an agnostic cloud reference architecture that respects ISO 27001 architecture controls.
- Comparison of three Cloud Services Providers in terms of native security services and tools and respectively identify which service comply each ISO control.
- Implement the agnostic Reference architecture in one Cloud Service Provider and do a proof of concept of the infrastructure and some services and tools.
- As the controls were based on annex A of ISO 27001, if in the future the objective is to be ISO compliant, this work can speed up this process and provide guidance for compliancy.

STATE OF THE ART

2.1 CyberSecurity OverView

2.1.1 What is CyberSecurity

Cybersecurity refers to a set of technologies, procedures, and practices designed to protect networks, devices, programs, and data against attack, damage, or unauthorized access. It can be divided into a few categories, **Network** security, **Application** security, **Information** Security, **Operational** security, **Disaster recovery and business continuity**, and **End-user education**.

- **Network** security is the process of protecting systems networks from intruders whether targeted attackers or some kind of malware.
- **Application** security aims to prevent data or code within the app from being stolen or hijacked and keeping the software and device free of threats. It addresses the security considerations that arise during the development and design of software, but also provides systems and methods to secure apps after they are deployed.
- **Information** Security protects the integrity and privacy of data, both in storage and in transit.
- **Operational** security are all the processes and decisions that are made to handle and protect data assets
- **Disaster recovery and business continuity** define how an organization responds to a cyber-security incident or any other event that causes the loss of operations or data. In order to return to the same operational capability as before the incident, disaster recovery policies determine how the company recovers its activities and records. Business continuity is the plan the organization falls back on while trying to operate without certain resources.

- **End-user education** human error is a major point of weakness, to counter that companies need to educate employees that they will be targeted, encouraging them to be vigilant at all times, teaching employees what qualifies as sensitive data, how to identify and avoid threats, acceptable use policies and security policies.

2.1.2 The importance of Cyber Security

Our society is more technologically reliant than ever before and there is no sign that this trend will slow down. We rely on computer systems every day, whether you are an individual, a small company or a large multinational. Personal data that could result in identity theft is now posted to the public on our social media. Sensitive information like credit card information and bank account details are now stored in our devices or in the cloud (Bruijn and Janssen, 2017).

With this society more dependent on technology, cybercrime is rising and cyber attacks are being more and more often and sophisticated, as a result, governments around the world are bringing more attention to cybercrimes. GDPR¹ is a great example, it has increased the reputational damage of data breaches by forcing all organizations that operate in the EU to communicate data breaches, appoints a data-protection, require user consent to process information, and anonymize data for privacy. A security breach or an successful attack can damage a business in a range of ways including Economic, Reputational and Regulatory costs.

Cyber attacks are **real and often**, as it is possible to verify in figure 2.1, regardless of the company's size and reputation they have to be ready and take security very seriously.

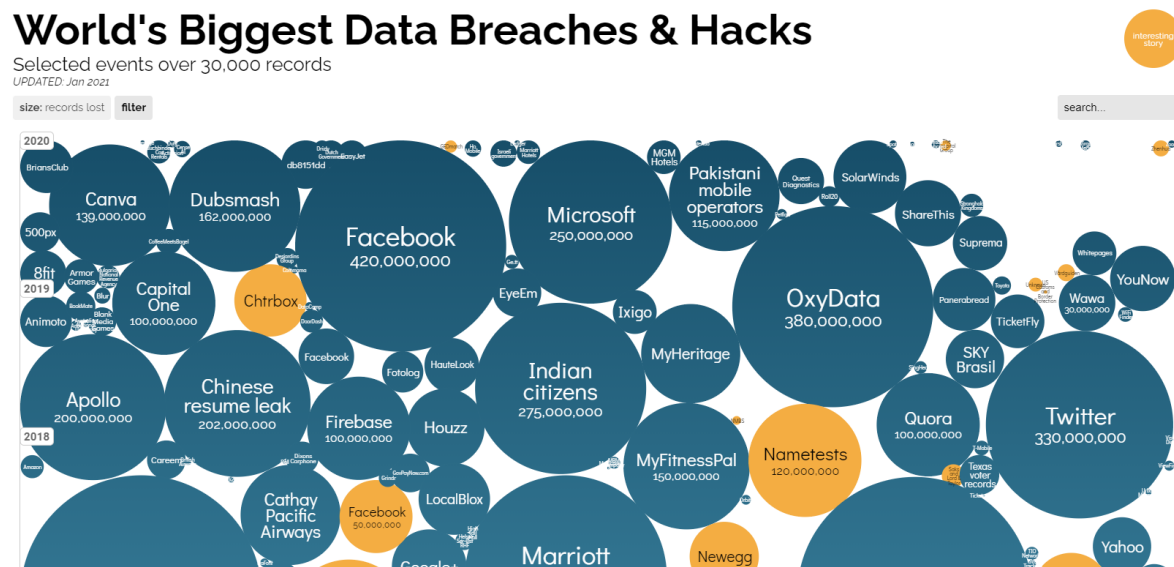


Figure 2.1: World's Biggest Data Breaches & Hacks

source by : <https://informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/>

¹<https://gdpr-info.eu/>

2.1.3 Security by Design

"Security is a process, not a product" - (Schneier, 2000) it isn't simply a technological problem but a processual issue that necessitates using the right tools to secure a solution's whole delivery chain ensuring that security is built in from the start - by design. The main goal and idea are that solutions need to be designed right from the beginning to be safe and secure.

Security by design focuses on preventing a breach rather than repairing the issue and restoring systems after a company has been hit by an attack. It enables companies to design and automate their environments with reliably coded security and governance also allows extending their cybersecurity capabilities for real-time governance, risk, and compliance reporting. In terms of bug fixing costs the later they're detected in the software development lifecycle, the higher the cost for fixing them. The principle of approaching security from the beginning to the end of a product development lifecycle is known as "Shift Left Security".

Implementing adequate measures, architecture designs, and security testing in the early development lifecycle, saves effort, time, and money to fix security issues.

2.1.4 Zero trust and Least privilege access

The principle of least privilege (PoLP), also known as the principle of minimal privilege or the principle of least authority, refers to an information security concept in which a user is given just only the necessary levels of access or permissions needed to perform his/her job functions. It is considered a security best practice and it is fundamental to protect access to sensitive data and assets. This principle should not only be applied to human users but to applications, systems, machines, and devices that require privileges or permissions to perform a required task.

The principle of least privilege ensures that a human or non-human has the requisite access needed and nothing more by that it reduces the attack surface, since one the common attacks are exploitation of privileged credentials, it prevents and stops the spread of malware and facilitate auditing and compliance. [(Schneider, 2003),(Dong, 2011)]

2.1.5 Defense in Depth

Defense in Depth is a cybersecurity strategy that layers several defensive techniques to protect sensitive data and information technology systems.

Organizations can use layered defense to decrease vulnerabilities, contain attacks, and manage risk. In basic terms, with a defense-in-depth strategy, if a bad actor breaches one layer the following layer of defense may be able to contain it. This approach should cover people, technology, and operations. It provides guidelines and best practices for securing physical infrastructure, organizational processes, and IT systems.(Reid, 2016)

2.2 Application Security

As described above, application security has some goals, but this section will be discussing how to test the Security of an Application.

2.2.1 The OWASP Testing Project

The Open Web Application Security Project ²(OWASP) is a nonprofit foundation that works to improve the security of software and produces freely-available articles, methodologies, documentation, tools, and technologies. The OWASP Testing Project has been in development for many years. The aim of the project is to help people understand the what, why, when, where, and how of testing web applications. The project has delivered a complete testing framework, not merely a simple checklist or prescription of issues that should be addressed. Readers can use this framework as a template to build their own testing programs or to qualify other people's processes. (OWASP, [WSTG-stable-INFO-02](#))

What is testing ? "Testing is a process of comparing the state of a system or application against a set of criteria". The problem is when people test against a set of mental criteria that are neither well defined nor complete and that result in numerous flaws

Why Perform Testing ? Allows organizations to compare themselves against industry peers, to understand the magnitude of resources required to test and maintain software and understanding the gap between existing practices and industry best practices.

When to Test ? Improve the Software Development Life Cycle (SDLC) by including security in each of its phases.

What to Test ? Test people, to ensure that there is adequate education and awareness. Process to ensure that there are adequate policies and standards and that people know how to follow these policies. Technology to ensure that the process has been effective in its implementation.

2.2.2 Testing Techniques

2.2.2.1 Manual Inspections and Reviews

Manual inspection are human evaluations that generally test the security consequences of people, policies, and processes. Inspection of technology decisions such as architectural designs can also include manual inspections. They are usually carried out by analyzing documentation or conducting interviews with the designers or system owners.

This can appear to be a simple concept but by asking someone how something works and why it was implemented in a specific way, the tester can quickly determine if any security concerns are likely to be evident.

²<https://owasp.org/>

2.2.2.2 Threat Modeling

Threat modeling has become a popular technique to help system designers think about the security threats that their systems and applications might face. Therefore, threat modeling can be seen as a risk assessment for applications. It enables the designer to develop mitigation strategies for potential vulnerabilities and helps them focus their inevitably limited resources and attention on the parts of the system that most require it. It enables to find at an early stage possible vulnerabilities and threats and implement strategies to mitigate them.

There are five steps to create a threat model:

- Decomposing the application
- Defining and classifying the assets
- Exploring potential vulnerabilities
- Exploring potential threat
- Creating mitigation strategies

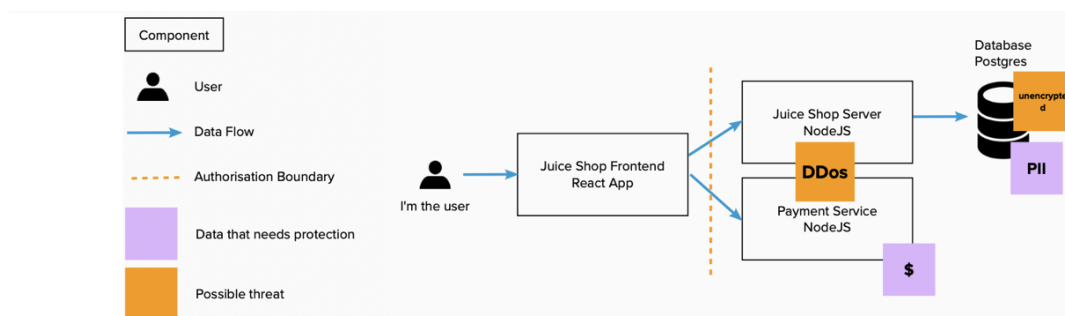


Figure 2.2: Threat Model example

2.2.2.3 Code Review

Source code review is the process of manually checking the source code of a web application for security issues. Many serious security vulnerabilities cannot be detected with any other form of analysis or testing.

All the information for identifying security problems is there in the code, somewhere!

Although there is nothing to replace this process of manual code review, there are some tools that can assist us.

- **Static application security testing (SAST)** - Is a testing methodology that analyzes the source code to find security vulnerabilities. SAST scans an application before the code is compiled, gives feedback in realtime.

- **Dynamic Application Security Testing (DAST)** - Involves examining the app during runtime. Usually doesn't provide the information that static analysis provides, but it is a good way to detect interesting elements from a user's point of view. Find security vulnerabilities or weak spots in a program while it is running.

It is clear that a SAST and a DAST have different goals, so they should complement each other and never choose between one or the other.

2.2.2.4 Penetration Testing

Penetration test or commonly known as ethical hacking is essentially the "art" of testing a system or application remotely to find security vulnerabilities. Typically, the penetration test team is able to access an application as if they were users. The tester acts like an attacker and attempts to find and exploit these vulnerabilities.

There are three types of testing, Black Box Testing where the tester impersonates the attacker and doesn't have any knowledge about the system. White Box Testing where the tester has full knowledge of the system like design, specification, documentation, and so on. Gray Box Testing where the tester has partial knowledge of the system.

2.2.3 OWASP Top 10

The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications. Using the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces more secure code (OWASP, 2017).

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

Figure 2.3: OWASP Top 10 2017

2.3 Container Security

2.3.1 Virtualization

"Virtualization has been in use for many years, but it is best known for enabling cloud computing. In cloud environments, hardware virtualization is used to run many instances of operating systems (OSs) on a single physical server while keeping each instance separate. This allows more efficient use of hardware and supports multi-tenancy- Souppaya et al., 2017

In figure 2.4 we can see the timeline of virtualization.

In traditional architectures (pre-virtualization), the operating system is directly installed on hardware devices, the operating system can only allocate the physical CPU and memory resources.

A hypervisor, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing. Hypervisors support the creation and management of virtual machines (VMs) by abstracting a computer's software from its hardware. Hypervisors make virtualization possible by translating requests between the physical and virtual resources.

"A **container** is a virtual runtime environment that runs on top of a single operating system (OS) kernel and emulates an operating system rather than the underlying hardware.

A **container** engine is a managed environment for deploying containerized applications. The container engine allocates cores and memory to containers, enforces spatial isolation and security, and provides scalability by enabling the addition of containers.- Firesmith, 2017

Unlike non-virtualization or hypervisor virtualization, the container virtualization **isolates** each application from each other, and with it goes their dependencies.

It is possible to combine hypervisor virtualization with container virtualization, where each VM has their own OS and each OS has the container engine, this process results in a hybrid architecture virtualization.

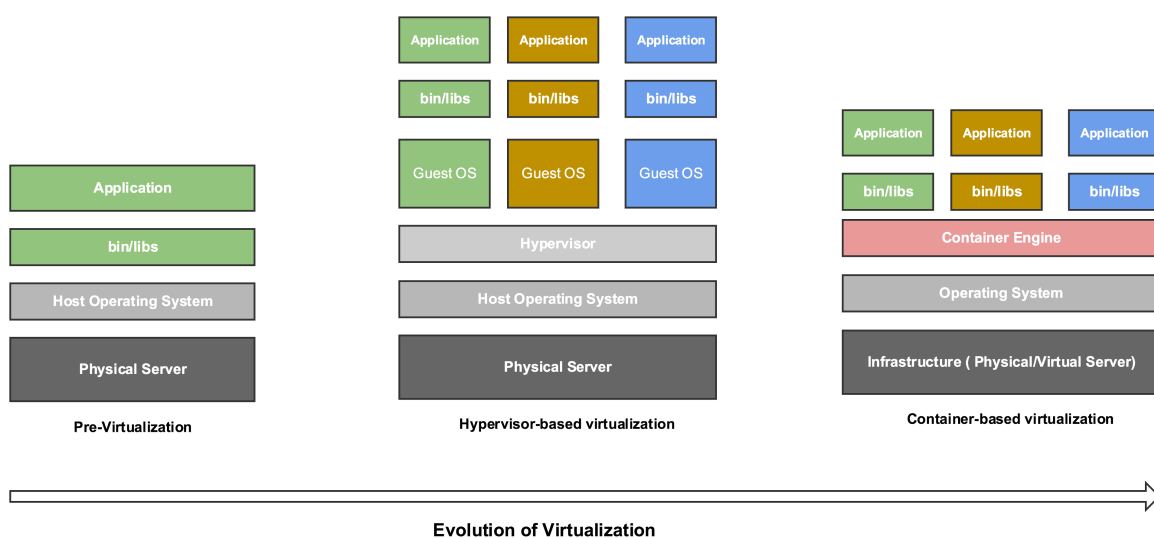


Figure 2.4: Virtualization

2.3.2 Major risk of containers technologies

Based on a typical container architecture represented in figure 2.5 we can detect what and where are the major risk of this technology, these risks are represented in figure 2.6.

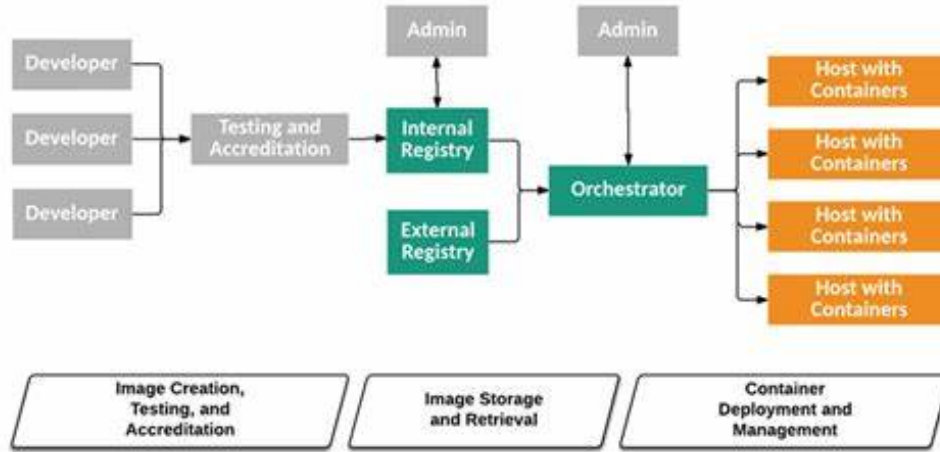


Figure 2.5: Container Architecture

	Images	Images Registry	Orchestrator	Container runtime	Host OS
Risks	<ul style="list-style-type: none"> • Images Vulnerabilites • Images Configuration • Embedded malware • Embedded clear text secrets • Use untrusted images 	<ul style="list-style-type: none"> • Insecure connections to registries • Stale images in registries • Insufficient authentication and authorization 	<ul style="list-style-type: none"> • Unauthorized access • Poorly separated inter-container network traffic • Mixing of workload sensitivity levels • Orchestrator node trust 	<ul style="list-style-type: none"> • Vulnerabilities within the runtime software • Unbounded network access from containers • Insecure container runtime configurations • App vulnerabilities • Rogue containers 	<ul style="list-style-type: none"> • Large attack surface • Shared kernel • Host OS component vulnerabilities • Improper user access rights • Host OS file system tampering

Figure 2.6: Major risk of containers technologies

2.4 DevSecOps

Security is just as critical to business success as the overall quality, performance, and usability of an application. As development cycles are shortened and delivery frequencies are increased, it becomes essential to ensure that quality and safety are built in from the very start.

DevSecOps is all about adding security to DevOps processes.

In figure 2.7 we can see that everything previously discuss should be implemented at a Software Development life cycle in order to develop more secure software and don't compromise either the Organization or their clients.

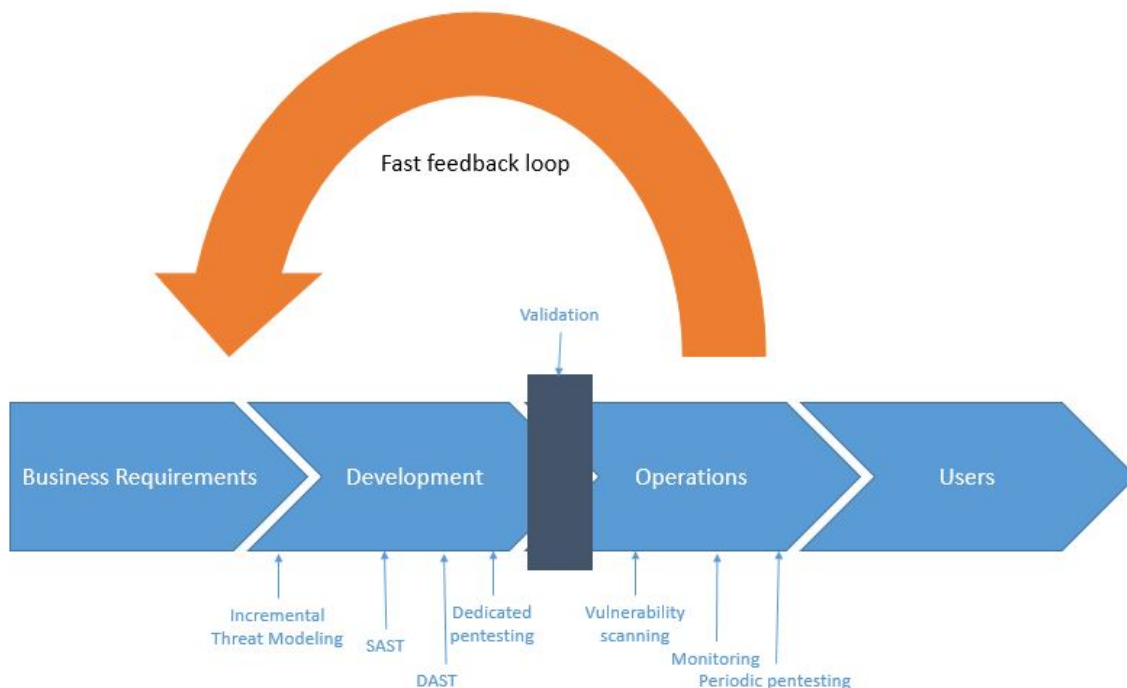


Figure 2.7: Example of DevSecOps

2.5 Cloud Shared Responsibility Model

The Shared Responsibility Model defines the distribution of responsibilities for security in the cloud between the cloud provider and the customer. The responsibility is different according to the service provided. In the cloud, we can have three kinds of services, Infrastructure as a Service (*IaaS*), Platform as a Service (*Paas*), and Software as a Service (*SaaS*).

IaaS is fully self-service for accessing and monitoring computers, networking, storage, and other services. *IaaS* allows businesses to purchase resources on-demand and as-needed instead of having to buy the hardware outright, as the name says, the users have full control of their infrastructure.

PaaS provides cloud components to certain software while being used mainly for applications. PaaS delivers a framework for developers that they can build upon and use to create customized applications. All servers, storage, and networking are managed by the cloud provider or a third-party provider while the developers maintain and manage their applications.

Software as a Service, also known as cloud application services, represents the most commonly utilized option for businesses. All the responsibilities are from the cloud service provider.

In figure 2.8 we can see the Shared Responsibility Model where are defined the users and cloud provider responsibilities of each service.

On-Premises	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Customer data	Customer data	Customer data	Customer data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
Operating system	Operating system	Operating system	Operating system
Virtualization	Virtualization	Virtualization	Virtualization
Networking	Networking	Networking	Networking
Storage	Storage	Storage	Storage
Servers	Servers	Servers	Servers

User/consumer responsibility
 Provider responsibility

Figure 2.8: Shared Responsibility Model

source by : <https://blogs.vmware.com/cloud/2021/06/10/improve-cloud-security-posture-management/>

2.6 ISO

ISO is an independent, non-governmental organization, that promote worldwide proprietary, industrial, and commercial standards. These standards aid in the creation of products and services that are safe, reliable, and of good quality and, help businesses increase productivity while minimizing errors and waste. They also serve to safeguard consumers and the end-users of products and services, ensuring that certified products conform to the minimum standards set internationally.

2.6.1 ISO 27001-Information Security Management

ISO 27001 Provides requirements for an information security management system (ISMS), organizations that accomplish ISO 27001 demonstrate to regulatory authorities that they take the security of information very seriously and, having identified the risks, done as much as is reasonably possible to address them. ISO 27001 has a document with a list of controls and their objectives named "Annex a", which is composed of 14 controls, where which control have their own sub controls in a total of 114 (ISMS, 2020). These controls are:

- A.5 – Information Security Policies
- A.6 – Organisation of Information Security
- A.7 – Human Resource Security
- A.8 – Asset Management
- A.9 – Access Control
- A.10 – Cryptography
- A.11 – Physical & Environmental Security
- A.12 – Operations Security
- A.13 – Communications Security
- A.14 – System Acquisition, Development & Maintenance
- A.15 – Supplier Relationships
- A.16 – Information Security Incident Management
- A.17 – Information Security Aspects of Business Continuity Management
- A.18 – Compliance

In order to understand what a system needs to fulfill to achieve ISO 27001 certificate, all 114 controls were analyzed and are described in figure 2.9. Once the goal is to shift these controls to the cloud, all controls related to physical security were ignored because that responsibility will be to the cloud service providers.

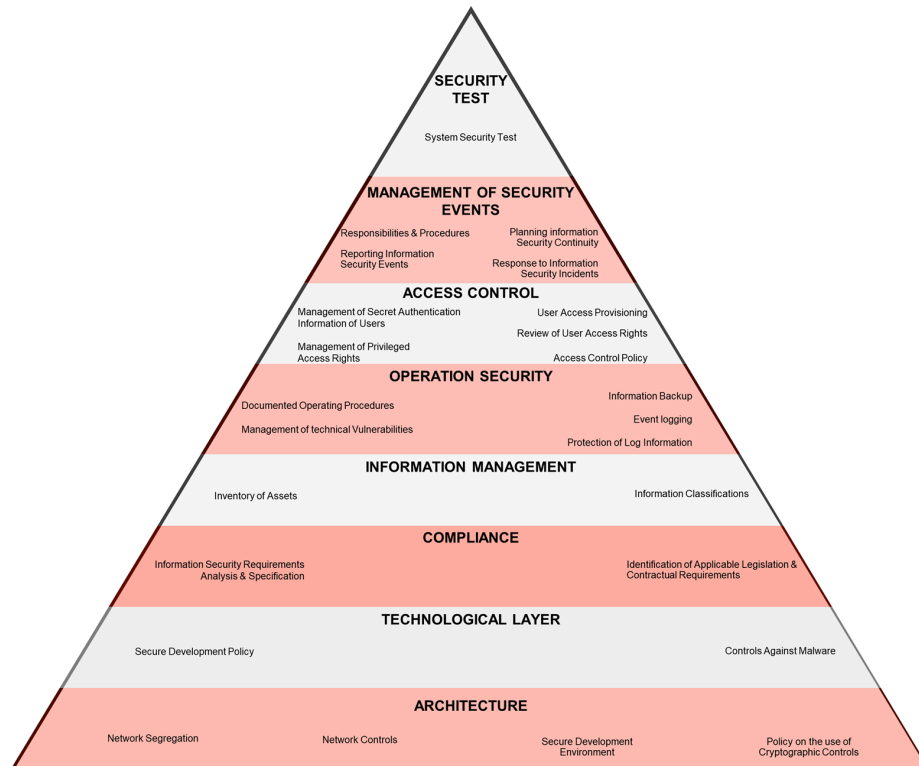


Figure 2.9: Annex A controls

Based on the study of the controls it was decided to divide the pyramid of 2.9 into 8 layers, **Architecture, Technological layer, Compliance, Information Management, Operation Security, Access Control, Management of Security Events**, and **Security Test**.

- **Architecture**

- *Network Segregation* - The architecture must implement the "Security Zoning", this means that we divide the architecture into zones, usually, the zones are the demilitarized zone (DMZ), Presentation zone, Application zone, DataBase zone, and Infrastructure zone.
- *Network Controls* - Refers to all network controls that should be implemented and configure properly, like Firewalls, Intrusion Detection Systems (IDS) that analyze network traffic for signatures that match known cyberattacks, Intrusion Prevention Systems (IPS) that also analyzes packets, but can also stop the packet from being delivered based on what kind of attacks it detects, Communications protocols, and so on.
- *Secure Development Environment* - Has the goal to protect and secure development environments for system development and integration efforts that cover the entire system development lifecycle.
- *Policy on the use of Cryptographic Controls* - Everything related to cryptographic are inserted in this control like which cipher is used, show the cryptographic fluxes, and how the database is encrypted.

- **Technological layer**

- *Secure Development Policy* - This refers to Operating Systems hardening techniques, Vulnerability management of the operating system, and application patches.
- *Controls Against Malware* - Implementation, configuration, and management of anti-virus

- **Compliance**

- *Information Security Requirements Analysis & Specification* -
- *Identification of Applicable Legislation & Contractual Requirements* - Organizations must be aware of all the regulations that their system, service, or application needs to respect are subjected to. Also, this control refers that the organizations must be contractually safeguard of any third-party that the system relies on.

- **Information Management**

- *Inventory of Assets* - Any assets associated with information and information processing facilities need to be identified and managed over the lifecycle, always up to date. A register or inventory of those assets has to be put together that shows how they are managed and controlled, based around their importance.
- *Information Classifications* - Information must be classified in terms of legal requirements, value, criticality and sensitivity to any unauthorised disclosure or modification, ideally classified to reflect business activity rather than inhibit or complicate it.

- **Operation Security**

- *Documented Operating Procedures* - Operating procedures must be documented and then made available to all users who need them. Documented operating procedures help to ensure consistent and effective operation of systems for new staff or changing resources.
- *Management of Technical Vulnerabilities* - Detailed management of technical vulnerabilities i.e. API, libraries and dependencies.
- *Information Backup* - Ensure that organizations implement, manage properly backups and data loss prevention technics.
- *Event Logging* - Event logs recording user activities, exceptions, faults and information security events need to be produced, kept and reviewed regularly. Logging and monitoring mechanisms form an important part of a “defence-in-depth” strategy for security management by providing both detective and investigation capabilities.
- *Protection of Log Information* - Logging facilities and log information must be protected against tampering and unauthorised access. It is also critical to ensure logs are stored in a secure and tamper-proof manner so that any evidence derived from them can be evidenced in a provable manner. This is especially important in any form of legal proceedings relating to evidence from the log.

- **Access Control**

- *Management of Secret Authentication Information of Users* - Secret authentication information is a gateway to access valuable assets. It typically includes passwords, encryption keys etc. so needs to be controlled through a formal management process and needs to be kept confidential to the user.
- *Management of Privileged Access Rights* - Managing usually more powerful and higher ‘privileged’ levels of access e.g. systems administration permissions versus normal user rights. The allocation and use of privileged access rights has to be tightly controlled.
- *User Access Provisioning* - A process (however simple and documented) must be implemented to assign or revoke access rights for all user types to all systems and services.
- *Review of User Access Rights* - Asset owners must review users’ access rights at regular intervals, both around individual change (on-boarding, change of role and exit) as well broader audits of the systems access. Authorisations for privileged access rights should be reviewed at more frequent intervals given their higher risk nature.
- *Access Control Policy* - An access control policy must be established, documented and reviewed regularly taking into account the requirements of the business for the assets in scope. Access control rules, rights and restrictions along with the depth of the controls used should reflect the information security risks around the information and the organisation’s appetite

for managing them. Put simply access control is about who needs to know, who needs to use and how much they get access to.

- **Management of Security Events**

- *Responsibilities & Procedures* - Describes how management establish responsibilities and procedures in order to ensure a quick, effective and orderly response to address weaknesses, events and security incidents. In simple terms an incident is where some form of loss has occurred around confidentiality, integrity or availability.
- *Reporting Information Security Events* - A good control here ensures that information security incidents and events can be reported through suitable management channels as soon as possible. Employees and associated interested parties (e.g. suppliers) need to be made aware of their obligations to report security incidents and you should cover that off as part of your general awareness and training
- *Planning Information Security Continuity* - The organisation must determine its requirements for information security and the continuity of information security management in adverse situations, e.g. during a crisis or disaste.
- *Response to Information Security Incidents* - It is always good to assign owners, be clear on actions and timescales. The individual placed in charge of dealing with the security event will be responsible for restoring a normal level of security.

- **Security Test**

- *System Security Test* - Performing all kinds of technics to test the system security, for example, pen testing, source code analysis, and threat model.

REFERENCE ARCHITECTURE

This chapter will show all the processes to develop a secure Cloud architecture considering all the ISO 27001 controls previously described. It will be compared three Cloud Service providers to choose the best one in terms of security and native controls that fulfill the ISO controls. (Newcombe, 2012)

3.1 On-Premise Reference Architecture

Before we move to the cloud, first was applied all the knowledge, good practices, and all the possible controls described in figure 2.9, to an On-premise environment. This approach reduces the possible misconfiguration, insecurity design, and gap between controls.

A consolidated and secure architecture on-premise will be easier, more efficient, and error-proof to migrate to a cloud base architecture.

The infrastructure must support an application on a multi-tier architecture with three tiers, the presentation tier (frontend), Application tier (backend), and Database tier, (IBM, 2020).

The application must be able to connect by mobile or Desktop clients via internet. Besides, there has to be an internal application for employees that aren't exposed to the internet.

Since part of the infrastructure will be exposed to the internet there is a risk associated with that. The application can be in sight of potential threats actors, like hackers or script kids. Other than attacks coming from the internet it also has to be prepared for inside organization attacks. Inside organization attacks are common and as referenced previously, we must adopt a defense-in-depth approach to get all the layers of security as possible.

Taking into account the risk and the ISO controls, the figure 3.1 represents the on-Premise reference Architecture

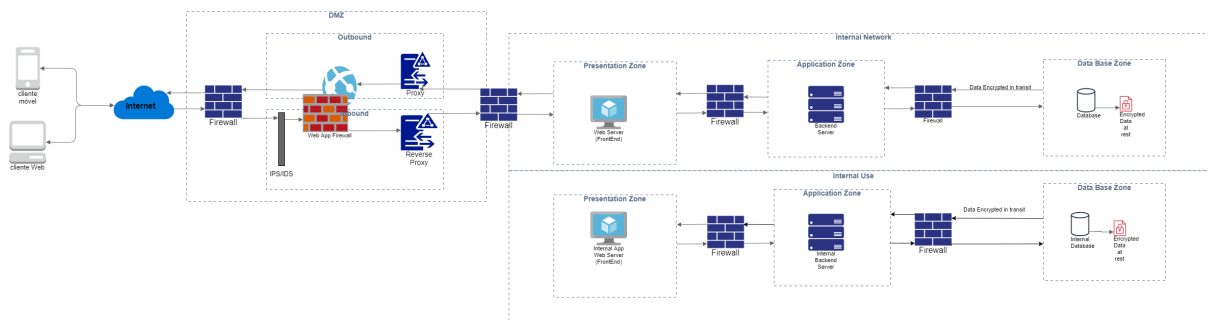


Figure 3.1: On-Premise Reference Architecture

3.1.1 Controls applied

The ISO 27001 controls represented in figure 3.1 are :

- *Network Segregation* -
 - Both applications have the three-tier zone physical segregated.
 - The application that is exposed to the internet has a demilitarized zone (DMZ) as the first line of defense to control North-South traffic which is the movement of data packets that are initially entering a network from the outside.
 - The internal application is segregated from the one exposed to the internet to prevent attacks from the internet, and in case of exposure the internal application is safeguarded.
- *Network Controls*
 - Firewalls between zones must implement whitelist rules such as an Access Control List (ACL) to make sure that each zone communicates only with the necessary zone(s) with that its possible to control the East-West traffic, which is the network traffic among devices or resources.
 - The DMZ must implement a network firewall to prevent OSI Network layer attacks, a web application firewall (WAF) and IDS and IPS to Application layer attacks, and proxy and reverse proxy to in case of an attack be possible to easily turn off the income and/or outcome respectively traffic to or from the application.
 - Communication protocol with Transport Layer Security(TLS) that is a widely adopted security protocol designed to facilitate privacy and data security for communications over the Internet
- *Policy on the use of Cryptographic Controls*
 - Use strong and not breakable cryptographic techniques and keys
 - Database encryption
 - Data encryption in transit

As mentioned before the objective is to develop a secure and reliable cloud architecture. The On-Premise architecture is a middle step, so wasn't done an intensive study and the architecture has not been fully developed to include the remaining controls. The focus was the architecture layer controls.

3.2 Cloud Service Provider agnostic architectures

Now that the On-Premise is developed, the next step is to migrate it to the cloud. The strategy adopted is first to create a Cloud Service Provider (CSP) agnostic architecture, thus allowing not to be restricted by only a CSP tools and services.

It will be focus on the architecture controls (2.9), just like the On-Premise architecture (3.1), but this time with general cloud tools and properties.

The application has the same requisites as the On-Premise.

3.2.1 Single VPC architecture

The first step is to create a network, inside of the cloud, that we have full control of, where the applications will be deployed, and ensure that is segregated from other cloud environments.

To assure full control and cloud segregation, from other environments, it's possible to configure a Virtual Private Cloud (VPC). A Virtual Private Cloud is a **secure, isolated private cloud hosted within a public cloud**. A VPC can be used to launch resources into a virtual network established, like run code, store data, host websites among others. This virtual network closely resembles a typical network that runs in a traditional data center, but with the added benefit of cloud-based scalability. Besides isolation, VPC offers other advantages like **Scalability**, since is hosted by a public cloud provider is possible to add more computing resources on-demand, **Easy hybrid cloud deployment**, since is simple to connect a VPC to an On-premise environment and **Better performance**. [(AWS, 2021c),(Cloudflare, 2021c)]

Now let's take all of the architecture controls from the on-premise architecture and move them to the cloud.

Network Segregation :

To have both applications segregated by the three-tier zones it will be used subnets. A subnet, or subnetwork, is a network inside a network or in this case is a network inside the previously set VPC. A subnet allows it to set a range of IPs, in order to, o split a large network into a grouping of smaller, interconnected networks to help minimize traffic. This segregation reduces network-wide threats by quarantining compromised sections of the network and by making it more difficult for trespassers to move around an organization's network. There are three types of subnets, *public subnet* where the traffic is routed to an internet gateway, *private subnet* if it doesn't have a route to the internet gateway, and *VPN-only subnet* if it doesn't have a route to the internet gateway, but has its traffic routed to a virtual private gateway for a VPN connection. Just like On-Premise, it will be created a DMZ subnet, Presentation Subnet, Application

subnet, and DataBase Subnet. [(Cloudflare, 2021b),(AWS, 2021b)]

Network Controls :

With the subnets deployed now, it's time to control the East-West traffic, for that will be used Network Security Groups (*NSG*) as the replacement of the between zones firewalls that are placed On-Premise architecture. A Network Security Group or Security Group acts like a firewall where allows it to filter network traffic to and from resources and virtual networks. It is possible to define rules to deny or allow specific traffic and set an Access Control List that has that purpose. For each rule, it is possible to specify source and destination, port, and protocol. The *NSG* will assure that the subnets only allow traffic from the desired subnet, this is, the Presentation subnet only accepts traffic from DMZ and Application subnet, the Application subnet only allows traffic from the Presentation and the DataBase subnets, and the Database subnet only allows traffic from the Application subnet. [(Azure, 2021),(Annamalai, 2014),(AWS, 2021a)]

Instead of a traditional network firewall, a valid optional is a Next-Generation Firewall (*NGFW*). A traditional firewall provides stateful inspection of network traffic. It allows or blocks traffic on OSI layer 3 and filters traffic based on administrator-defined rules. A next-generation firewall does this, and so much more, like packet filtering, Stateful inspection, and VPN awareness. In addition to access control, NGFWs can block modern threats such as advanced malware and application-layer attacks, it can be viewed as an IDS/IPS add-on and not a replacement. (Cloudflare, 2021a)

There is no generic cloud tool or service for IDS/IPS, WAF, proxy, and reverse proxy, so these controls will be reliant on the CSP. However, these controls will be represented in the agnostic architecture.

The communication protocol must still be HTTPS.

For this architecture, it was added a VPN connection to the internal application with the purpose of internal use for employees.

To obtain the many advantages of the cloud, like scalability and availability, a Network Load Balancer (*NLB*) will be placed, before the DMZ. A Network Load Balance automatically distributes the incoming traffic originating from the internet across multiple targets. This increases the availability of the application because it can handle millions of requests per second. The *NLB* distributes traffic across the registered targets in its subnet only. In short, the Network Load Balancer distributes network load efficiently across multiple servers, ensures high availability and reliability by sending requests only to servers that are online and, provides the flexibility to add or subtract servers as demand dictates. (Google, 2021b)

Behind each subnet will be an Application Load Balancer. This has the same goal as the Network load balance but for distributing the traffic for each server by demand. With *ALB* it is possible to automatically scale the resources by demand.

Policy on the use of Cryptographic Controls

Like some elements of the DMZ, the cryptographic controls depends on the CSP, although most CSP has default data encryption at rest and in movement, there isn't an agnostic tool or service. So to clarify all data flows must be encrypted and use strong cryptographic protocols and technics. In the same way that

these controls aren't represented in the On-Premise architecture, they won't be in the agnostic architecture as well.

The agnostic architecture is defined in figure 3.2, there is possible to see the controls represented and the first shift of On-Premise architecture to the cloud.

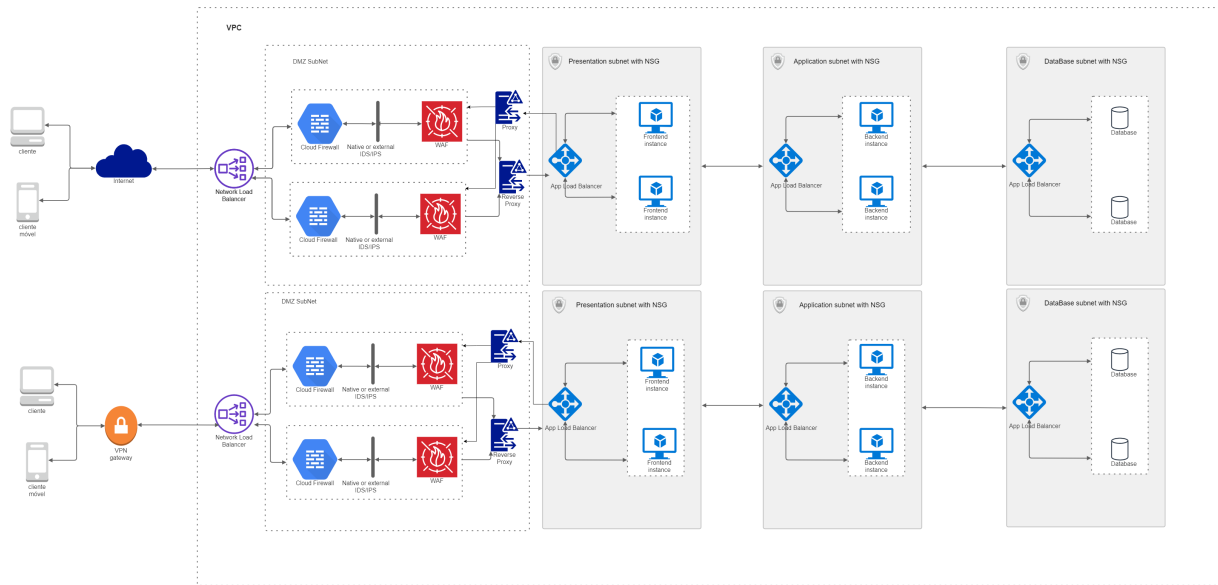


Figure 3.2: Single VPC Reference Architecture

3.2.2 Multi VPC architecture

Until now it was presented two architectures, besides the controls implemented and requisites both architectures have a unique environment. Now all the changes will be on the cloud architecture, where is easier to expand the infrastructure, and for that, it was added a new requisite, **the infrastructure must be able to support multiple environments**, for example, Development environment, Pre-Production environment, and Production environment. In addition to being an ISO 27001 control, 2.9, it is a regular practice of Organizations, so the cloud architecture must ensure that the organization can deploy, securely, their environments.

To accomplish this requisite, the Single-VPC agonist architecture will improve to a Multi-VPC architecture, where each environment will be deployed in each own VPC. This strategy increases the segregation and consequently the security of each environment, especially the Development environment where is more propitious to have vulnerabilities and misconfigurations. "Using a Multi-VPC architecture allows you to isolate different parts of your infrastructure.- (Wittig, 2016)

Besides, the agnostic architecture, in figure 3.2, lacks some resources efficiency on the DMZ subnet. That means that any application has its own DMZ subnet, if we scale the numbers of applications it means that the numbers of DMZ subnets will grow as well. So the infrastructure will have repeated resources for the same function. Once the DMZ is the first line of defense the rules, filter, protection, and all other controls of the DMZ are the same for all applications in the infrastructure. So instead of having the DMZ multiple times replicated, it is possible to remove it from a subnet and places it on a VPC that will be shared by all environments and distribute the traffic to the respective application.

As mentioned before, each environment will be deployed in each own VPC, for simplification, it will be deployed three environments, the Development, the Production, and an Internal one for the employee's application. That means three VPC, and an extra one for the DMZ what makes a total of four. With this improvement comes a challenge, which is, how will the VPC communicate. It has to be considered three factors, scalability, performance, and configuration. There are mainly two types of connections, the **VPC-to-VPC peering**, and a **Transit Gateway**.(AWS, 2020)

Represented in figure 3.3 is a VPC-to-VPC peering design, this approach is a direct forward connection between two VPCs. Is a simple way to connect two VPCs, enables full bidirectional connectivity between the VPCs. Although is not the most efficient in terms of scalability, for example, if there is VPC peering between VPC A and VPC B and VPC A to VPC C, traffic from B to C cannot travel through A to C, there must be a direct connection from B to C. This means with a growth infrastructure, resources, and instances, there will be more VPC peering connections, therefore, harder to configure and maintain those connections, besides, usually there is a limitation of VPC peering connections per VPC.

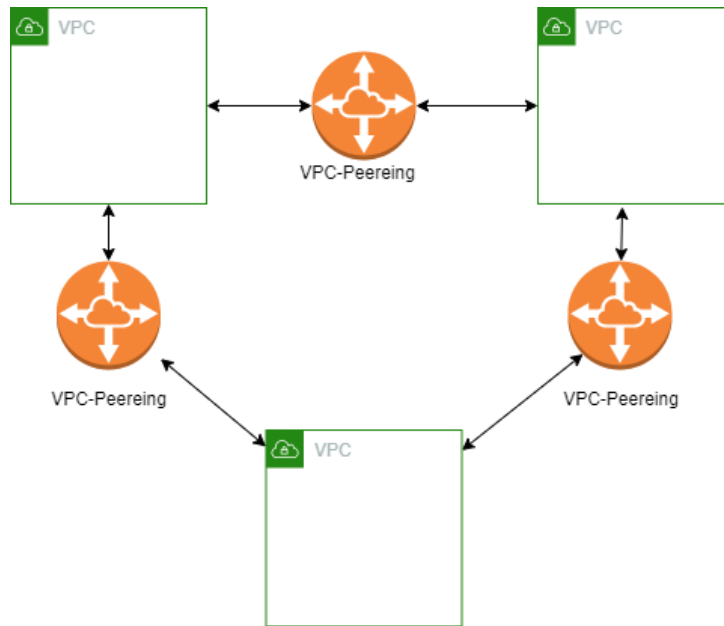


Figure 3.3: VPC-to-VPC Peering Design

Transit Gateway is like a hub and spoke where a gateway receives all the traffic and redirect to the respective VPC. Transit Gateway enables customers to connect thousands of VPCs, with this strategy is easier to scale infrastructure with multiple VPCs. Usually, the Transit Gateway comes with a higher cost than a VPC peering, the cost is related to the number of connections and, GB transferred. Figure 3.4 shows how the Transit Gateway design is.

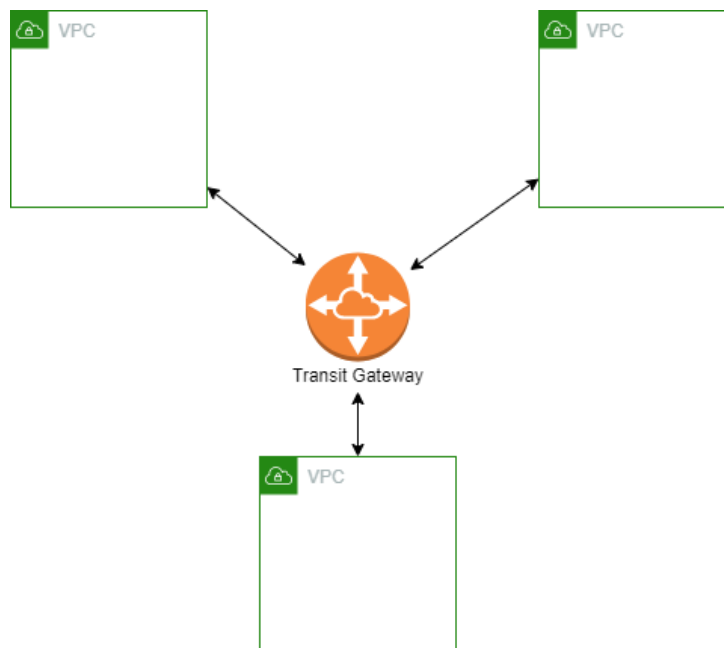


Figure 3.4: Transit VPC Design

For the reference architecture was chosen the transit gateway as the VPC communication approach, since the infrastructure must handle multiple environments, the main criteria are scalability and easy configuration.

The last control added is the event and centralization of the logs. To improve the architecture was added a logging appliance with a central repository to have full control of what is happening in all environments. This event and centralization of logs are an ISO control mentioned on Operation Security layer of figure 2.9.

The figure 3.5 is the Multi-VPC reference architecture in an agnostic solution where all controls above are represented.

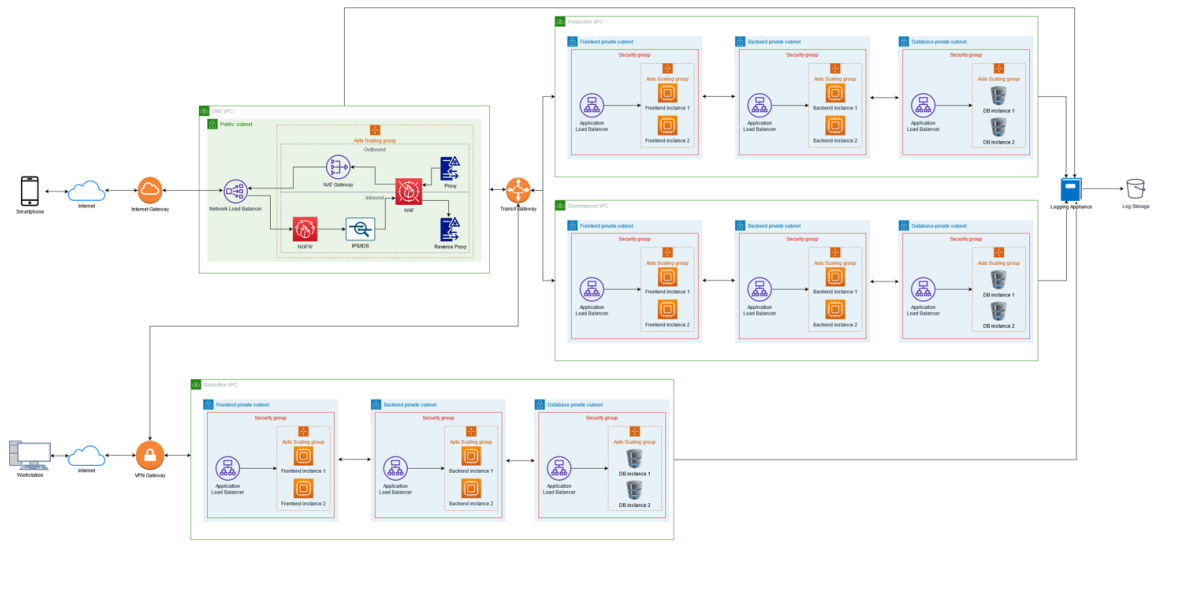


Figure 3.5: Multi- VPC reference Architecture

3.3 Cloud Service Providers

The next step is to choose a Cloud Service Provider to materialize the agnostic architecture. According to c-sharpcorner¹ and Gartner², the most Cloud Service Providers for infrastructure as code are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), therefore, these will be compared to decide which one will implement the architecture. It will have three criteria to compare these CPS, the first and the most important is the Native Security controls that assure the ISO requirements. The Second is the maturity of the CPS, this means, the community, documentation, and support. The last one is the cost, based on the Architecture which CSP is cheaper to implement.

The documentation of each CSP was analyzed, in order to, find the native service or tool that assures the ISO controls represented on each layer of pyramid 2.9. That comparison is represented in table 3.1 where for each layer and for each control the native service that assures the control of each CSP are represented. In that table, it is possible to see that GCP has a few gaps in comparison to the other two, a lot of that because of its late entrant, to the IaaS market. Between AWS and Azure, both assure the same controls besides the Azure that doesn't have an Incident Secure Response service.(Google, 2021a)

Since the AWS is older and has a lot more clients and users than the rest of the others, it is natural that the community and documentation are more complete. While the native controls were studied and analyzed it was possible to verify that the AWS has more community support and documentation than the Azure and GCP.

In terms of the cost, AWS is the most expensive, followed by Azure and the one cheaper is the GCP.(Solanki, 2021)

Although the GCP is the cheapest CSP, the principal goal and criteria is Security, so the priority is the native controls, and for that reason, GCP will not be an option to materialize the infrastructure. Since AWS and Azure have practically the same controls covered, besides the one already mentioned, the key differentiation was the documentation and community support which AWS has more advantage in this field.

Taking into account everything mentioned above the Cloud Service Provider chosen to materialize Architecture 3.5 will be the Amazon Web Services.

¹<https://www.c-sharpcorner.com/article/top-10-cloud-service-providers/>

²<https://www.gartner.com/technology/media-products/reprints/AWS/1-271W10T3-PTB.html>

Native Security Controls				
Layer	ISO Control	AWS	Azure	GCP
Architecture	Network Segregation	VPC,Subnet	VPC,Subne	VPC,Subnet
	Network Controls	AWS Network Firewall,Shield,AWS WAF	Azure Fire-wall,Azure WAF	Google Cloud Firewall,Cloud IDS,Google Cloud Armor
	Secure Development Environment	VPC,Subnet	VPC,Subnet	VPC,Subnet
	Policy on the use of Cryptographic Controls	AWS Encryption SDK	Azure Storage Service Encryption (SSE)	GCP default Encryption
Technological layer	Secure Development Policy	AWS Systems Manager Patch Manager, CIS Hardened Images	CIS Hardened Images	CIS Hardened Images
	Controls Against Malware	Shield	Microsoft Antimalware	Google Cloud Armor Managed Protection
Compliance	Information Security Requirements Analysis & Specification	AWS Inspector	n/d	n/d
	Identification of Applicable Legislation & Contractual Requirements	Artifact	Service Trust Portal	n/d
Information Management	Inventory of Assets	AWS Config	Azure Security Control	Cloud Asset Inventory
	Information Classifications	AWS IAM	Azure Identity Management	Identity and Access Management
Operation Security	Documented Operating Procedures	n/d	n/d	n/d
	Management of Technical Vulnerabilities	Security Hub	Security Center	Security Command Center
	Information Backup	AWS Backup, Glacier	Backup	n/d
	Event Logging	Cloud Trail, Cloud Watch	Azure Monitor	Cloud Logging
	Protection of Log Information	Cloud trail, Cloud watch	Azure Audit Logs	Cloud Audit Logs

Native Security Controls				
Access Control	Management of Secret Authentication Information of Users	AWS KMS	Key vault	Secret Manager
	Management of Privileged Access Rights	IAM	Azure Identity Management	Identity and Access Management
	User Access Provisioning	IAM	Azure Identity Management	Identity and Access Management
	Review of User Access Rights	IAM	Azure Identity Management	Identity and Access Management
	Access Control Policy	IAM	Azure Identity Management	Identity and Access Management
Management of Security Events	Responsibilities & Procedures	n/d	n/d	n/d
	Reporting Information Security Events	n/d	n/d	n/d
	Planning Information Security Continuity	AWS Backup, CloudEndure	BackUp, Site Recovery	n/d
	Response to Information Security Incidents	AWS Detective	n/d	n/d
Security Test	System Security Test	n/d	n/d	n/d

n/d - Not Defined

Table 3.1: CSP Native Security Controls.

PROOF OF CONCEPT

This chapter has the goal to implement and test the infrastructure in AWS. For the POC will be deployed a dummy application. The next sections will be explained the AWS architecture, as well as the application and a walkthrough of all AWS services and controls.

The dummy application will be a page with checkboxes where someone can do a security assessment of their system. The technologic layer is ReactJs for the frontend, Java spring boot for the backend, and H2 engine for the database. The backend loads the controls stored in the database and exposes an API that will be consumed by the frontend.

4.1 AWS POC Architecture

Now that the CPS is selected is time to convert the agnostic services into AWS services.

To test the infrastructure hosting the dummy application there was no need to implement a second environment and the internal application. The controls applied to a single environment and a single application should be replicated and used with a multi-environment and applications. Since the only challenge is the configuration of the tools and services to support those.

Figure 4.1 represents the AWS POC architecture, the following is a walkthrough of the AWS services and controls.

- **DMZ :**
 - **Network Firewall** - AWS doesn't provide a native next-generation firewall, so was opted for the traditional NetWork Firewall.
 - **AWS Shields** - AWS Shield is a managed Distributed Denial of Service (DDoS) protection service, although there is not an IDS/IPS native service, the AWS Shields can integrate with WAF and intercept and identify potential attacks.

- **AWS WAF** - Besides the protection that any WAF does of layer 7, the AWS WAF helps protect web applications or APIs against bots that may affect availability, compromise security, or consume excessive resources.
 - **Proxy and Reverse Proxy** - There is not a native service that acts as a proxy, so the solution is to instantiate machines with a proxy service like Nginx.
 - **AWS CloudFront** - Delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the request is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance. It can be integrated with security tools.
 - **S3 VPC endpoint** - Has a policy that controls the use of the endpoint to access Amazon S3 resources where the frontend will be deployed.
- **DashBoard VPC :**
 - **BackEnd Instance** - The backend instance that will allocate the server will be an EC2.
 - **Database Instance** - The Database instance that will allocate the database will be an AWS RDS.
 - **API Gateway** - AWS API Gateway is a fully managed service that makes it easy to create, publish, maintain, monitor, and secure APIs at any scale.
 - **Serverless Services :**
 - **React FrontEnd Component** - The frontend will be deployed on AWS s3 that is an object storage service offering industry-leading scalability, data availability, security, and performance.
 - **Log Storage** - Like the frontend the logs will be saved on a dedicated S3, that only will store logs.
 - **CloudWatch** - This is a monitoring service for resources and applications. CloudWatch collects and tracks metrics, collects and monitors log files, sets alarms, and automatically reacts to changes in Amazon Web Services resources.
 - **CloudTrail** - This is a service that enables governance, compliance, and operational and risk auditing of AWS accounts. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs. It's possible also to view, search, download, archive, analyze, and respond to account activity across your AWS infrastructure. It identifies who or what took which action, what resources were acted upon when the event occurred, and other details to help analyze and respond to activity in AWS accounts.
 - **AWS Key Management Service** - Is a managed service that makes it easy to create and control the encryption keys used to encrypt data, allows to manage secrets and keys.

- **Aws Infrastructure Services**

- **GuardDuty** - Is a threat detection service that continuously monitors your AWS accounts and workloads for malicious activity and delivers detailed security findings for visibility and remediation. Analyzes and processes the VPC Flow Logs, AWS CloudTrail management event logs, CloudTrail S3 data event logs, and DNS logs.
- **Inspector** - Is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Automatically assesses applications for exposure, vulnerabilities, and deviations from best practices. Can automate security vulnerability assessments.
- **Detective** - Automatically collects log data from AWS resources and uses machine learning, statistical analysis, and graph theory to build a linked set of data that enables you to easily conduct faster and more efficient security investigations.
- **Security Hub** - Is a service that allows to aggregate security alerts from other tools or services in a single place. Analyze security trends and identify the highest priority security issues and best practices. Security hub improve the visibility of AWS infrastructure.
- **Macie** - Automates the discovery of sensitive data, such as personally identifiable information (PII) and financial data, to provide a better understanding of the data that is stored in S3 buckets. It is a control of Data Loss Prevention that also provides an inventory of S3 buckets, and it automatically evaluates and monitors those buckets for security and access control.
- **Identity and Access Management** - Is a service that helps to securely control access to resources. IAM control who is authenticated (signed in) and authorized (has permissions) to use resources.

- **AWS Account Services**

- **Organizations** - is an account management service that enables the consolidation of multiple AWS accounts into an organization. includes account management and consolidated billing capabilities.
- **AWS Management Console** - is a web application that comprises and refers to a broad collection of service consoles for managing Amazon Web Services. This control allows using all other services that aren't represented on the architecture flow but are but are mentioned in table 3.1.

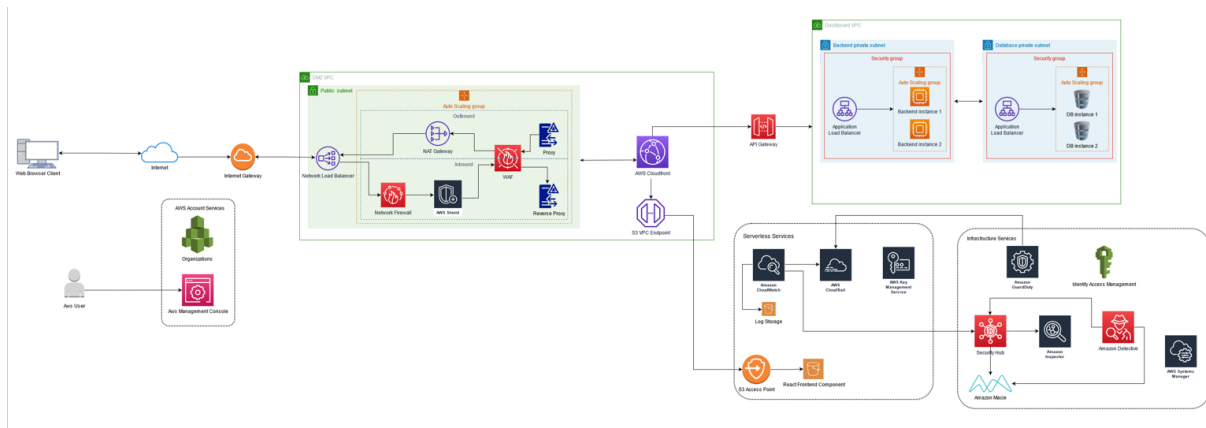


Figure 4.1: AWS Reference Architecture

As explained before, the architecture will not have a multi-environment, so will be used two VPC, one to deploy the DMZ and the other to deploy the Dashboard of the dummy application.

The DMZ is composed of a Network Load Balancer, a Network Firewall, AWS Shield, Web Application Firewall, a proxy and reverse proxy, and an AWS CloudFront. The functionality of the load balancer, Network Firewall, WAF, proxy and reverse proxy has already been described in the Agnostic Architecture chapter. The AWS shield will protect against DDOS attacks that will safeguard applications running on AWS. Uses techniques such as deterministic packet filtering and priority-based traffic shaping to mitigate basic network and transport layer attacks automatically. CloudFront is the “front door” to the application and infrastructure. The primary attack surface is moved away from critical content, data, code. CloudFront, together with Shield and WAF creates a flexible, layered security perimeter against multiple types of attacks. These controls together provide a scalable, reliable, and high-performance security perimeter.

The dashboard VPC where the application will be deployed will have the Backend Subnet and Database Subnet, representing the application zone and database zone, respectively. The presentation zone will be allocated in a serverless service, the AWS S3 bucket. A serverless is a cloud-native development model that allows developers to build and run applications without managing servers. There are still servers in serverless, but they are abstracted away from app development. In the dashboard VPC, the two subnets have the security groups that will act as a firewall to apply network traffic rules just like the agnostic architecture. The API gateway will expose and serve the API from the application backend. As mentioned, the presentation zone will be deployed in an S3 bucket. It will be using Identity and Access Management (IAM) to ensure that the bucket is private and not open to direct access from the internet. Also, with IAM, we can force the bucket only to accept traffic from the CloudFront through the S3 VPC endpoint. We assure that the traffic has to pass through the DMZ with this policy.

With Identity and access management, policies will be created to define which user or service has permission to access what or do what action. With this service, we can assure the least privileged access.

The CloudWatch will collect logs and track metrics, variables of the resources, and applications. It also allows configuring triggers based on alarms or metrics. CloudWatch gains system-wide visibility into resource utilization, application performance, and operational health. This bucket will have to IAM policies

to assure which services are allowed to use or access the logs.

CloudTrail has a more account focus, it collects logs and tracks the activity of user account, creating a trail or history of each user behaviour. It gains visibility into the AWS accounts activity by viewing, searching, downloading, archiving, analyzing, and responding to account activity across AWS infrastructure. It can identify who or what took which action, what resources were acted upon when the event occurred, and other details.

AWS Key Management Service will be used to create and manage AWS keys. These keys will be for the logs and database encryptions. It will be set an automatic rotation of the cryptographic material and delete KMS keys to complete the key lifecycle.

As mentioned before, GuardDuty continuously monitors and analyzes logs to identify unexpected and potentially unauthorized and malicious activity. It enables the identification of potential escalations of privileges, uses of exposed credentials, or communication with malicious IP addresses or domains. In addition, this can detect unauthorized infrastructure deployments, like instances deployed in a Region that has never been used, or unusual API calls, like a password policy change to reduce password strength.

AWS Security Hub is a centralized security alert. It can integrate other services to centralize logs, alerts, or alarms, allowing it to have more visibility of AWS infrastructure in a single place.

AWS Inspector will produce a detailed list of security findings prioritized by severity level. These findings can be reviewed directly or as part of detailed assessment reports. AWS Inspector helps find vulnerabilities to mitigate them.

AWS Detective uses machine learning to automatically extracts time-based events such as login attempts, API calls, and network traffic from AWS CloudTrail and Amazon VPC flow logs. It also consumes findings detected by GuardDuty. AWS Detective can rapidly investigate any activity that falls outside the norm, identify patterns that may indicate a security issue, and understand all the resources affected by a finding.

AWS Systems Manager is more a strand of resources management and configuration. This service allows performing patches, scale, view, investigate, and resolve operational work. This easy configuration can mitigate some misconfigurations and speed up the patching to fix a known vulnerability. Integrating with IAM, we assure only administrator accounts have the permission to use the Systems Manager.

AWS Organizations is to manage accounts into an organization or infrastructure where contains the log archive account, the audit account, and the resources they own.

All the other services that aren't represented on the architecture flow will be aggregated on the AWS management console, where it is possible to set up, deploy, and configure those. For example, AWS Backup, CloudEndure, Glacier, and Config

4.1.1 Implementation

To deploy the reference architecture was used the paradigm Infrastructure as code (*IaC*). IaC is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. With this strategy, the infrastructure deployment and maintenance is faster, consistent, efficiently and decreases the probability of miss configurations. (Microsoft, 2021)

Terraform¹ is an infrastructure-as-code tool that enables to build, update, and version infrastructure securely and efficiently. Terraform provide documentation² to set, deploy and configure the AWS services and resources. This tool will be used to materialize the AWS reference architecture 4.1.

Until here, pricing or any costs was not a criterion or a limitation, but now that we want to implement the architecture developed, that is a huge problem. Most of the services described and mentioned in architecture 4.1 have associated costs, so we need to adapt the architecture only to use free tier services because of lack of budget.

Architecture 4.2 is the final version of the POC infrastructure, already adapted with only free tier AWS services.

It is possible to check that the component that suffered the most impact was the DMZ. Most of DMZ services are paid, only AWS WAF, and Shield has a free version but with reduced functionalities. Free tier WAF only offers bots protection, so attacks of the Application layer like SQL injection or XSS that are on top of the OWASP top 10 aren't blocked. The Shield will protect against the most common, frequently occurring DDoS attacks. So the DMZ now will be the Cloudfront act like a proxy and reverse proxy, connected with WAF and Shield Standard of the free tier.

The Dashboard VPC has almost no impact, only the resources set of the EC2 and DataBase servers, like Ram, Memory, and CPU, are limited to the free tier usage. Was used the port 22 SSH with server-side authentication just to deploy the dummy application backend as the first instance, and then the port was closed. The front end has no impacts, and the policy of only accepting traffic from the CloudFront is the same.

Some services were removed because they didn't offer any free tier functionalities. These services are AWS Macie, Network Firewall, Config, Backup, Organizations and cloud endure.

Some Services have free trial days like Guarduty, Inspector, Security Hub, and Detective. The other services have free tier usage but with limitations or minor functionalities.

With this huge impact, we re-arrange the layer where the services are. The identity and access management now are on AWS Account Services because, as administrator, I will define the permissions policy. The KMS moved from ServerLess services to AWS services. Although KMS is a serverless service, it makes more sense to be part of the infrastructure services because KMS will be directly used by the administrator and not just running by himself.

¹<https://www.terraform.io/>

²<https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

Of course, this budget limitation will raise the risk and reduce the protection. Indeed, we can't guarantee all the ISO controls with the free tier. We have to consider this when performing a security test of the infrastructure in the future.

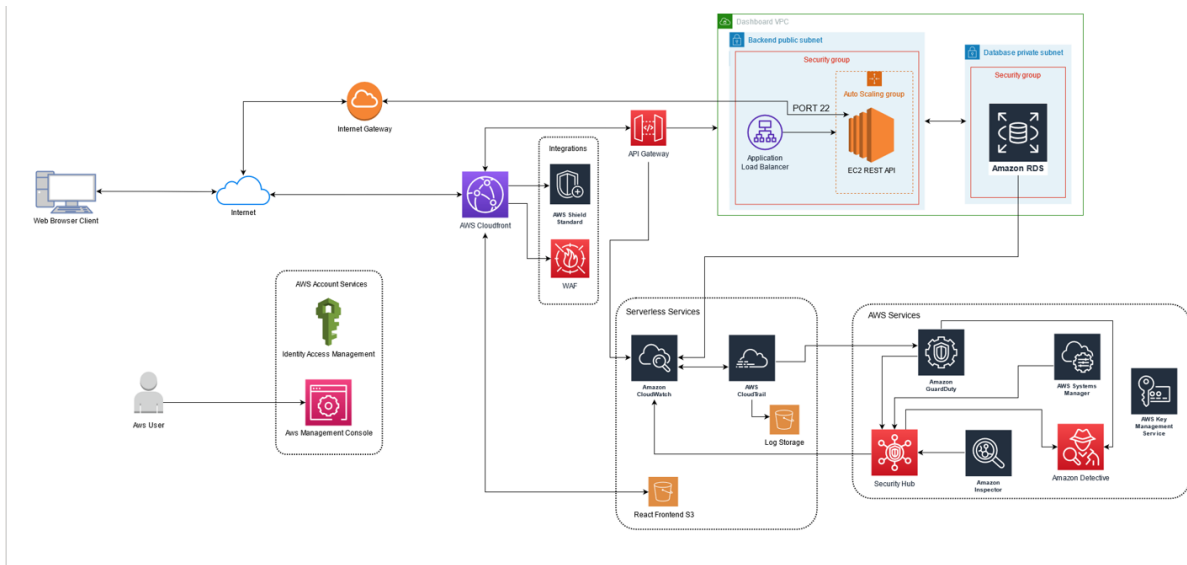


Figure 4.2: POC Architecture

Below are some examples of the terraform code to deploy and configure some instances or services implemented.

The next segment represents the IAM policy where we force the traffic of the S3 bucket that hosts the front end only to accept traffic from the CloudFront.

```

1 data "aws_iam_policy_document" "s3_iam_doc" {
2   statement {
3     actions = ["s3:GetObject"]
4     resources = ["${aws_s3_bucket.react_bucket.arn}/*"]
5
6     principals {
7       type = "AWS"
8       identifiers = [aws_cloudfront_origin_access_identity.cloudfront.iam_arn]
9     }
10  }
11
12  statement {
13    actions = ["s3:ListBucket"]
14    resources = [aws_s3_bucket.react_bucket.arn]
15
16    principals {
17      type = "AWS"
18      identifiers = [aws_cloudfront_origin_access_identity.cloudfront.iam_arn]
19    }
20  }
21 }

```

This one is the setup of the S3 bucket that stores the CloudTrail logs. We point to the bucket where the Cloudtrail will store the logs and use one symmetric Key from the Key Management Service to encrypt them.

```

1 resource "aws_cloudtrail" "cloudtrail" {
2   name = "cloudtrail"
3   s3_bucket_name = aws_s3_bucket.cloudtrail-bucket.id
4   s3_key_prefix = "CloudTrailLogs"
5   # include_global_service_events = false
6   kms_key_id = aws_kms_key.log_bucket_key.arn
7
8   event_selector {
9     read_write_type = "All"
10    include_management_events = true
11
12    data_resource {
13      type = "AWS::S3::Object"
14
15      values = ["${aws_s3_bucket.react_bucket.arn}/*"]

```

```
16 }
17 }
18 }
```

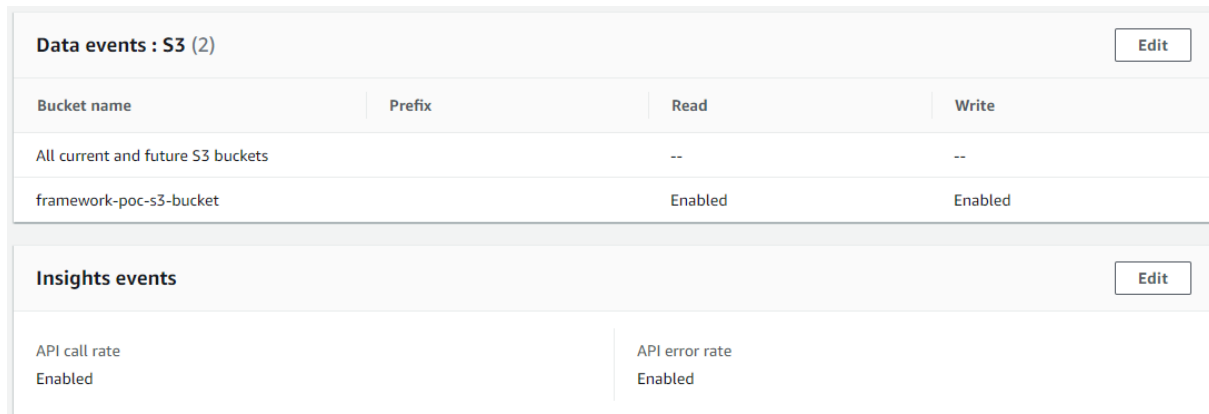
For last, this is a configuration example of the network security group of the Database subnet. We can see that the ingress and egress traffic are only to and from the backend subnet. So no other traffic is allowed to access the database.

```
1 # Database Security Group
2 resource "aws_security_group" "subnet-database-sg" {
3   name = "rds-sg"
4   vpc_id = aws_vpc.app-vpc.id
5
6   # Only MySQL in from Backend Security Group
7   ingress {
8     security_groups = [aws_security_group.subnet-backend-sg.id]
9     from_port = var.rds_port
10    to_port = var.rds_port
11    protocol = "tcp"
12    description = "MySQL"
13    //cidr_blocks = [var.subnet_backend_range]
14  }
15
16  # Only MySQL out to Backend Security Group
17  egress {
18    security_groups = [aws_security_group.subnet-backend-sg.id]
19    from_port = var.rds_port
20    to_port = var.rds_port
21    protocol = "tcp"
22    description = "MySQL"
23    //cidr_blocks = [var.subnet_backend_range]
24  }
25 }
```


4.2 Controls in Action

This section will show some services and controls in action in the AWS management console. Will present some configuration, alarms, dashboard, and information and findings that the implemented controls have of the infrastructure.

Figure 4.3 shows the trails configurations of the CloudTrail. We set the CloudTrail to monitor the front-end bucket, logs bucket, and API call and error rate. Cloud trail will monitor accounts behavior that affects these resources.

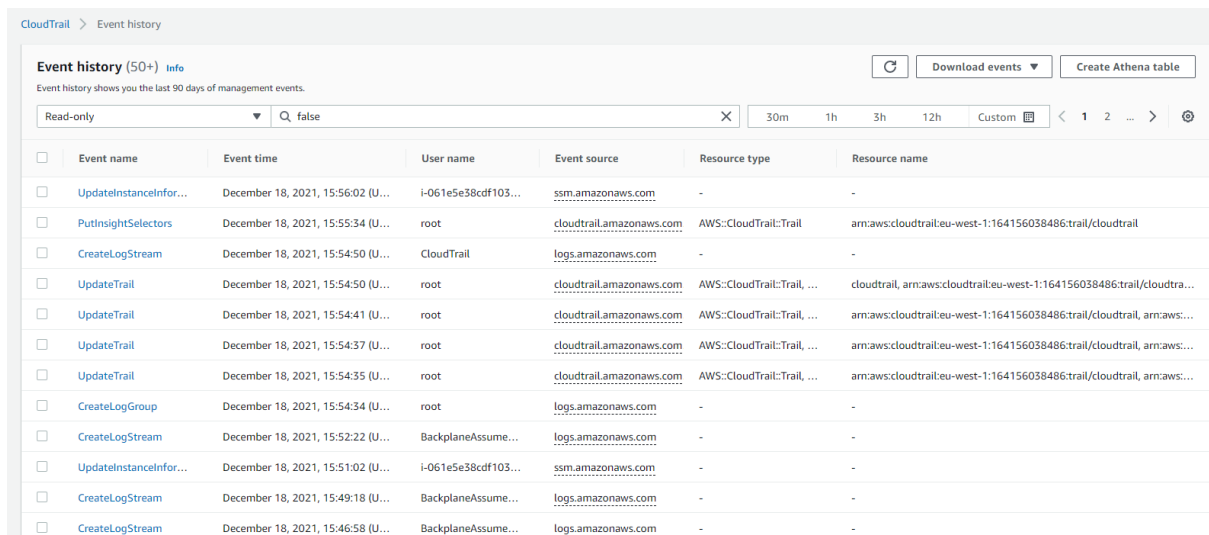


Data events : S3 (2) Edit			
Bucket name	Prefix	Read	Write
All current and future S3 buckets		--	--
framework-poc-s3-bucket		Enabled	Enabled

Insights events Edit	
API call rate Enabled	API error rate Enabled

Figure 4.3: CloudTrail trail configuration

In figure 4.4 are represented some logs collected by CloudTrail. We can see which user or service did what in what resource.



Event name	Event time	User name	Event source	Resource type	Resource name
UpdateInstanceInfor...	December 18, 2021, 15:56:02 (U...	i-061e5e38cdf103...	ssm.amazonaws.com	-	-
PutInsightSelectors	December 18, 2021, 15:55:34 (U...	root	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail	arn:aws:cloudtrail:eu-west-1:164156038486:trail/cloudtrail
CreateLogStream	December 18, 2021, 15:54:50 (U...	CloudTrail	logs.amazonaws.com	-	-
UpdateTrail	December 18, 2021, 15:54:50 (U...	root	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail, ...	cloudtrail, arn:aws:cloudtrail:eu-west-1:164156038486:trail/cloudtra...
UpdateTrail	December 18, 2021, 15:54:41 (U...	root	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail, ...	arn:aws:cloudtrail:eu-west-1:164156038486:trail/cloudtrail, arn:aws:...
UpdateTrail	December 18, 2021, 15:54:37 (U...	root	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail, ...	arn:aws:cloudtrail:eu-west-1:164156038486:trail/cloudtrail, arn:aws:...
UpdateTrail	December 18, 2021, 15:54:35 (U...	root	cloudtrail.amazonaws.com	AWS::CloudTrail::Trail, ...	arn:aws:cloudtrail:eu-west-1:164156038486:trail/cloudtrail, arn:aws:...
CreateLogGroup	December 18, 2021, 15:54:34 (U...	root	logs.amazonaws.com	-	-
CreateLogStream	December 18, 2021, 15:52:22 (U...	BackplaneAssume...	logs.amazonaws.com	-	-
UpdateInstanceInfor...	December 18, 2021, 15:51:02 (U...	i-061e5e38cdf103...	ssm.amazonaws.com	-	-
CreateLogStream	December 18, 2021, 15:49:18 (U...	BackplaneAssume...	logs.amazonaws.com	-	-
CreateLogStream	December 18, 2021, 15:46:58 (U...	BackplaneAssume...	logs.amazonaws.com	-	-

Figure 4.4: CloudTrail logs

As mentioned before, CloudWatch allows having more visibility in our resources. To test the CloudWatch alarms, we set two alarms. The first is to notify if the API get three or more forbidden request in one minute. The second is to report if we get five or more forbidden access in five minutes to the S3 front-end. This forbidden request refers to direct access to the front-end S3 bucket. As referred before, the front-end is

only accessible by the CloudFront. All other traffic is dropped. Figure 4.5 shows the CloudWatch alarms dashboard.

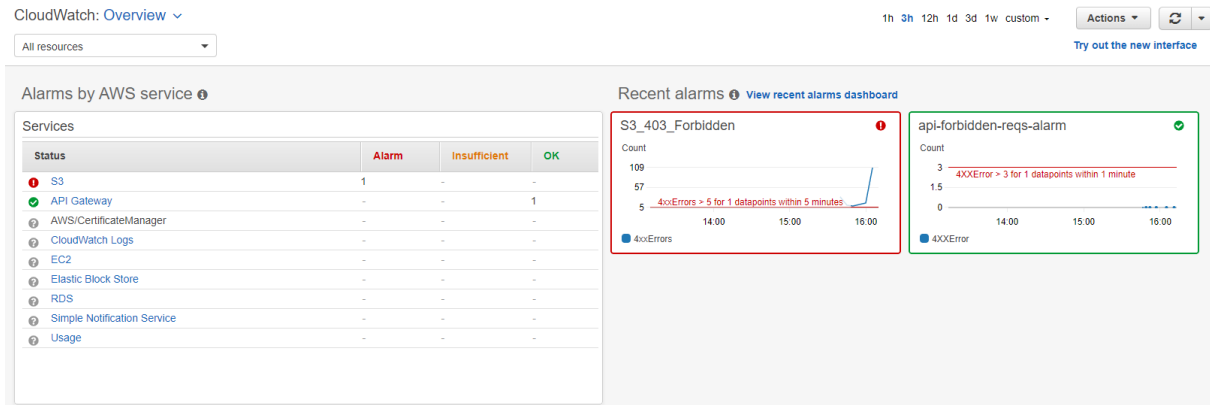


Figure 4.5: CloudWatch Dashboard

The following image 4.6 shows the notification email triggered by the CloudWatch alarm. It is possible to set a triggered response, like block or time out the Ip of who triggered the alarm.

```

Alarm Details:
- Name: S3_403_Forbidden
- Description:
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [6.0 (18/12/21 16:10:00)] was greater than the threshold (5.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Saturday 18 December, 2021 16:15:24 UTC
- AWS Account: 164156038486
- Alarm Arn: arn:aws:cloudwatch:eu-west-1:164156038486:alarm:S3_403_Forbidden

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanThreshold 5.0 for 300 seconds.

Monitored Metric:
- MetricNamespace: AWS/S3
- MetricName: 4xxErrors
- Dimensions: [BucketName = framework-poc-s3-bucket] [FilterId = EntireBucket]
- Period: 300 seconds
- Statistic: SampleCount
- Unit: not specified
- TreatMissingData: missing

State Change Actions:
- OK:
- ALARM: [arn:aws:sns:eu-west-1:164156038486:Default_CloudWatch_Alarms_Topic]

```

Figure 4.6: CloudWatch alarm notification via email

CloudWatch group logs by resource, and we can set the time of retention of the logs. Each organizations has to define the time to live of the logs analysing the risk and the needs of the logs. We can see that configuration in 4.7

<input type="checkbox"/>	Log group	Retention
<input type="checkbox"/>	/aws/apigateway/welcome	Never expire
<input type="checkbox"/>	/aws/rds/instance/terraform-20211217162512674700000002/error	Never expire
<input type="checkbox"/>	/aws/rds/instance/terraform-20211218142834128700000002/error	Never expire
<input type="checkbox"/>	API-Gateway-Execution-Logs_4vrh775jb/api	3 days
<input type="checkbox"/>	aws-cloudtrail-logs-164156038486-40c5ca56	Never expire

Figure 4.7: CloudWatch logs groups

Figure 4.8 shows the security findings aggregated in the Security hub. 4.8 shows the security findings aggregated in the Security hub.

<input type="checkbox"/>	Severity	Workflow status	Record State	Region	Company	Product	Title	Resource	Compliance Status	Updated at
<input type="checkbox"/>	LOW	NEW	ACTIVE	eu-west-1	AWS	Security Hub	2.6 Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket	Account 164156038486	WARNING	an hour ago
<input type="checkbox"/>	LOW	NEW	ACTIVE	eu-west-1	AWS	Security Hub	3.9 Ensure a log metric filter and alarm exist for AWS Config configuration changes	Account 164156038486	FAILED	an hour ago
<input type="checkbox"/>	LOW	NEW	ACTIVE	eu-west-1	AWS	Security Hub	3.8 Ensure a log metric filter and alarm exist for S3 bucket policy changes	Account 164156038486	FAILED	an hour ago
<input type="checkbox"/>	LOW	NEW	ACTIVE	eu-west-1	AWS	Security Hub	3.7 Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer created CMKs	Account 164156038486	FAILED	an hour ago
<input type="checkbox"/>	LOW	NEW	ACTIVE	eu-west-1	AWS	Security Hub	3.6 Ensure a log metric filter and alarm exist for AWS	Account 164156038486	FAILED	an hour ago

Figure 4.8: Security Hub dashboard

In the dashboard, the security hubs provide a brief summary of the vulnerabilities found by criticality. We can see that a critical vulnerability was found in the infrastructure. 4.9 represents the Security hub summary.

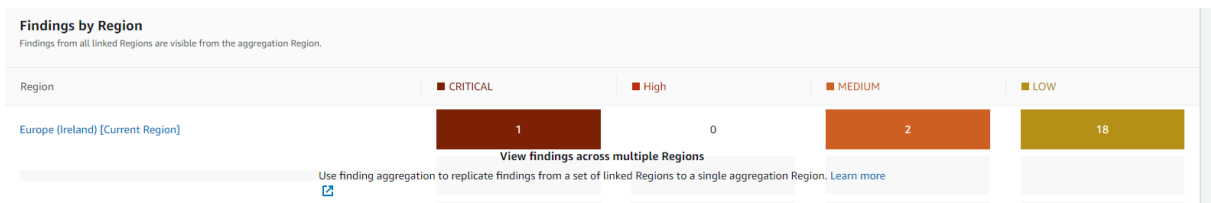


Figure 4.9: Security Hub summary

The critical vulnerability identified in the security hub is the S3 bucket logs used to store CloudTrail logs was with public access 4.10. This misconfiguration was mitigated by defining a correct policy on the bucket.

<input type="checkbox"/>	Severity	Workflow status	Record State	Region	Company	Product	Title	Resource	Compliance Status	Updated at
<input type="checkbox"/>	CRITICAL	NEW	ACTIVE	eu-west-1	AWS	Security Hub	2.3 Ensure the S3 bucket used to store CloudTrail logs is not publicly accessible	Account 164156038486	WARNING	an hour ago

Figure 4.10: Critical Vulnerability find by Security hub

The GuardDuty shows in 4.11 that the root privileges were used to do some action. And generate a report on what actions were and by what IP. This helps to prevent and find privileges escalation attacks.

The screenshot shows the AWS GuardDuty dashboard. A notification at the top states: "New feature: GuardDuty is now available in the AWS Asia Pacific (Osaka) Region. You can now extend your continuous security monitoring and threat detection to the AWS Asia Pacific (Osaka) Region. Learn more". The main content area displays a list of findings for the policy "IAMUser/RootCredentialUsage". Two findings are visible:

Finding type	Resource	Last detected	Count
Policy:IAMUser/RootCredentialUsage	Root: ASIASM0DWLVL634EYTF	6 minutes...	113
Policy:IAMUser/RootCredentialUsage	Root: ASIASM0DWLVLKB2PVBAS	26 minut...	16

The right-hand pane provides details for a selected finding. The title is "Policy:IAMUser/RootCredentialUsage" with finding ID "8cbe070aed08f0bc0b4792431f15654". The severity is "Low" and the region is "eu-west-1". The count is 113. The resource ID is "framework-poc-s3-bucket". The finding was created at "12-18-2021 15:07:59 (an hour ago)" and updated at "12-18-2021 16:20:00 (a few seconds ago)". The resource affected is "framework-poc-s3-bucket" with role "TARGET" and type "AccessKey".

Figure 4.11: GuarDuty dashboard

The KSM key dashboard in 4.12 presents the information of the keys. Information like their usage, status and type.

Aliases	Key ID	Status	Key spec	Key usage
-	489627fd-c077-420b-9b66-1ea314c04df4	Enabled	SYMMETRIC_DEFAULT	Encrypt and decrypt
-	e4e69099-b5d7-460c-986f-edcd58b44ed4	Pending deletion	SYMMETRIC_DEFAULT	Encrypt and decrypt

Figure 4.12: KMS keys dashboard

The AWS system manager in 4.13 shows how easy it is to apply patches to the infrastructure instances.

The screenshot shows the AWS Systems Manager Patch Manager dashboard. The main heading is "AWS Systems Manager Patch Manager" with the sub-heading "Automate patching with a native AWS solution". Below this, it states: "Centralized management for patching your fleet of Amazon EC2 Windows and Linux instances or your on-premises servers and virtual machines (VMs)."

The right-hand pane is titled "Patch your instances" and contains the following options:

- Patch instances without a schedule. [Patch now](#)
- Create schedules to patch instances. [Configure patching](#)
- Not ready to configure patching? Learn more about patching options by viewing the predefined patch baselines. [View predefined patch baselines](#)

Figure 4.13: AWS System Manager patch dashboard

CONCLUSION AND FUTURE WORK

This thesis has initially presented a synthesis of the ISO 27001 controls and an overview of cybersecurity in the industry. With that knowledge consolidated was developed two reference architecture, one On-Premise and another agnostic in the cloud. Further on we compare three cloud service providers in terms of native security tools and services, that comply with the ISO Controls, the documentation offered, and price. That comparison allows it to choose a Cloud Service Provider to materialize the agnostic reference architecture with the controls identified on the comparison previously effectuated. Was developed as well a dummy application to deploy in the infrastructure. Due to a limited budget, we have to adapt the reference architecture to only use the free tier services. In the end, we test the infrastructure services implementations and behavior.

We identified the native services and tools needed to respect the ISO controls on the three most popular cloud service providers. But the agnostic reference architecture allows it to materialize in any Cloud Service provider, missing only the mapping of the native services and tools to the ISO controls. With this work, we identify the controls, services, and tools that must be implemented and develop a safe and secure cloud infrastructure.

If one day the goal is to be ISO 27001 compliant, this study speeds up the process and benefits the organization because all the services, tools, and architecture were based in annex A ISO 27001.

In short, although the cloud has a large attack surface, with the right services and tools we can reduce this surface and keep hosted applications secure. Note that no system is unbreakable or impenetrable, we have to use all possible means and strategies to make our system as secure as possible.

Future work will be to try to obtain a budget to implement the infrastructure with all identified services and tools. And finally, do a blue team/red team exercise, where the red team goes through the attacker and tries to penetrate or damage the infrastructure and the blue team will monitor the infrastructure and apply responses to the attacks..

Bibliography

- Annamalai, N. (2014). *Network security groups*. <https://azure.microsoft.com/pt-pt/blog/network-security-groups/>
- AWS. (2020). Building a scalable and secure multi-vpc aws network infrastructure. <https://d1.awsstatic.com/whitepapers/building-a-scalable-and-secure-multi-vpc-aws-network-infrastructure.pdf>
- AWS. (2021a). *Security groups for your vpc*. https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html
- AWS. (2021b). *Vpcs and subnets?* https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Subnets.html
- AWS. (2021c). *What is amazon vpc?* <https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html>
- Azure. (2021). *Network security groups*. <https://docs.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview>
- Bruijn, & Janssen. (2017). Building cybersecurity awareness: The need for evidence-based framing strategies. 34, 1–7. <https://doi.org/https://doi.org/10.1016/j.giq.2017.02.007>
- Cloudflare. (2021a). *What is a next-generation firewall (ngfw)?* <https://www.cloudflare.com/learning/security/what-is-next-generation-firewall-ngfw/>
- Cloudflare. (2021b). *What is a subnet? | how subnetting works?* <https://www.cloudflare.com/learning/network-layer/what-is-a-subnet/>
- Cloudflare. (2021c). *What is a virtual private cloud (vpc)?* <https://www.cloudflare.com/learning/cloud/what-is-a-virtual-private-cloud/>
- Dong, X. M. R. L. Z. L. J. L. M. (2011). Specifying and enforcing the principle of least privilege in role-based access control. *Concurrency and Computation: Practice and Experience* vol. 23 iss. 12.
- Firesmith, D. (2017). *Virtualization via containers*. https://insights.sei.cmu.edu/sei_blog/2017/09/virtualization-via-containers.html
- Google. (2021a). *Compare aws and azure services to google cloud*. <https://cloud.google.com/free/docs/aws-azure-gcp-service-comparison>
- Google. (2021b). *External tcp/udp network load balancing overview*. <https://cloud.google.com/load-balancing/docs/network>
- IBM. (2020). *Three-tier architecture*. <https://www.ibm.com/cloud/learn/three-tier-architecture>
- ISMS. (2020). *Introducing annex a controls*. <https://www.isms.online/iso-27001/annex-a-controls/>

- Microsoft. (2021). *What is infrastructure as code?* <https://docs.microsoft.com/en-us/devops/deliver/what-is-infrastructure-as-code>
- Newcombe, L. (2012). Securing cloud services: A pragmatic approach to security architecture in the cloud.
- OWASP. (2017). *The owasp top ten*. <https://owasp.org/www-project-top-ten/>
- OWASP. (WSTG-stable-INFO-02). *The owasp testing project*. <https://owasp.org/www-project-web-security-testing-guide>
- Pettey, C. (2020). *Cloud shift impacts all it markets*. Retrieved October 26, 2020, from <https://www.gartner.com/smarterwithgartner/cloud-shift-impacts-all-it-markets/>
- Reid, G. (2016). *Detection in depth*. <https://blogs.cisco.com/security/detection-in-depth>
- Schneider, F. (2003). Least privilege and more. *IEEE Security Privacy Magazine* vol. 1 iss. 5.
- Schneier, B. (2000). The process of security. https://www.schneier.com/essays/archives/2000/04/the_process_of_secur.html
- Solanki, J. (2021). *Cloud pricing comparison 2021: Aws vs azure vs google cloud*. <https://www.simform.com/blog/compute-pricing-comparison-aws-azure-googlecloud/>
- Souppaya, M. P., Morello, J., & Scarfone, K. (2017). Application container security guide. <https://doi.org/https://doi.org/10.6028/NIST.SP.800-190>
- Wittig, A. (2016). Beyond the default: A multi-vpc architecture. <https://cloudonaut.io/beyond-the-default-a-multi-vpc-architecture/>

List of Figures

2.1	World's Biggest Data Breaches & Hacks	4
2.2	Threat Model example	7
2.3	OWASP Top 10 2017	9
2.4	Virtualization	10
2.5	Container Architecture	11
2.6	Major risk of containers technologies	11
2.7	Example of DevSecOps	12
2.8	Shared Responsibility Model	13
2.9	Annex A controls	15
3.1	On-Premise Reference Architecture	20
3.2	Single VPC Reference Architecture	23
3.3	VPC-to-VPC Peering Design	25
3.4	Transit VPC Design	25
3.5	Multi- VPC reference Architecture	26
4.1	AWS Reference Architecture	33
4.2	POC Architecture	36
4.3	CloudTrail trail configuration	39
4.4	CloudTrail logs	39
4.5	CloudWatch Dashboard	40
4.6	CloudWatch alarm notification via email	40
4.7	CloudWatch logs groups	41
4.8	Security Hub dashboard	41
4.9	Security Hub summary	41
4.10	Critical Vulnerability find by Security hub	41
4.11	GuarDuty dashboard	42
4.12	KMS keys dashboard	42
4.13	AWS System Manager patch dashboard	42