Expertise
and insight
for the future

Amir Nazarbeigi

Migration to cloud and security

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

27 March 2021

Author	Amir Nazarbeigi
Title	Migration to Cloud and security
Number of Pages	39 pages
Date	27 March 2021
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Mobile Solutions
Instructors	Kari Aaltonen, Senior Lecturer Hemmo Latvala, Head Developer TrueMed Oy

The purpose of this study is to explore the preparations needed when migrating company data to the cloud. In addition, how to secure the environment and the system from the beginning of the design is analyzed.

Resources used in this thesis consist of books, journals, articles and statistics regarding the subject. In addition, experiments conducted by the writer are discussed in the study. The thesis consists of a theoretical part which defines the concepts of cloud computing and the implementation part, which explains the technologies and their usage in the cloud migration system and the role they play when creating the foundation for the entire system.

As a result of this project, migration of data to the cloud commissioned by the owner of the project was successfully carried out. To ensure the quality, overall system reliability and security of this project, the entire process including the testing phase have been closely reviewed by stakeholders.

Keywords	Cloud, Data Security, Amazon Web Services, Encryption, KMS, Private Cloud

Contents

List of Abbreviations

Contents

List of Abbreviations

ORM	Object-relational mapping. The set of rules for mapping objects in a programming language to records in a relational database, and vice versa.
DBMS	Database management system. Software for maintaining, querying and updating data and metadata in a database.
CI	Continues integration
CD	Continues Deployment
SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
API	Application Programming Interface
OS	Operating System
GDPR	General Data Protection Regulation
VPN	Virtual Private Network
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
AZ	Availability Zone
EFS	Elastic File System
EBS	Elastic Block Storage
CPU	Central Processing Unit

POC	Proof of Concept
GPU	Graphics Processing Unit
HDD	Hard Disk Drive
KMS	Key Management System
AI	Artificial Intelligence
ML	Machine Learning
CDN	Content Delivery Network
CVS	Concurrent Versions System
UI	User Interface
CLI	Command-Line Interface
HC	HashiCorp
KV	Key Value
SSH	Secure Shell
S3	Simple Storage Service
FIPS 140-2	Federal Information Processing Standard Publication 140-2
VPC	Virtual Private Cloud
EC2	Elastic Compute Cloud
DoS	Denial of Service
UFW	Uncomplicated Firewall

IDS	Intrusion Detection System
TLS	Transport Layer Security
SSL	Secure Sockets Level
CA	Certificate Authority
RBAC	Role Based Access Control

1 Introduction

Cloud computing is a model for enabling clients with omnipresent, and on-demand access to shared pool of configurable computing resources that can be swiftly provisioned and released with minimal management overhead. (Mell & Grance, 2011)

In the recent years, more companies are realizing the importance of cloud computing and are adapting cloud in different ways, such as data storage, moving their IT infrastructure, etc. which gives them more freedom and requires less time-consuming management of resources, and configuration.

1.1 My thesis topic

In today's world the internet has become one of the important aspects of peoples' lives from connectivity to other people using social media to going to the doctor in 2020-2021 because of the coronavirus restrictions. With that in mind, many challenges will raise to be answered such as how can the application be able to grow as the number of the users grow how to ensure that their data is safe and not easily exposed for malicious use.

To answer these questions, cloud computing is becoming more and more popular with the providers offering many services for the companies and developers to easier create, test, and deploy their application as well as any other use cases it may relate to.

In this thesis the main goal is to analyze the preparations needed in order to migrate all the resources to the cloud and how to maximize privacy and security of the data from ground zero. To achieve this, this study starts by defining what is cloud and then moves on to choosing the providers and how to secure different part of the application from cloud, storage and application side.

2 Cloud computing

2.1 What is cloud computing?

The term “cloud computing” has gained many meanings throughout the years but at its core cloud computing is delivering computing services such as servers, storage, networking, etc. over the internet, on-demand, cost-efficient, in a remote location, without residing on owner’s computer, hand-held computational devices, or organizations own servers. (Wyld, 2009) There are many benefits in using cloud computing such as agility, elasticity and faster deployment.

Cloud services can help users, and organizations in different ways. Usage for organization A might be different from organization B. To best understand and help both of the organizations one should be familiar with different aspects of the cloud, such as characteristics, service models, capabilities, vulnerabilities, and deployment model to come about best solution and architecture.

2.2 Cloud computing characteristics

Based on NIST definition (Mell & Grance, 2011) cloud should have the following essential characteristics:

1. On-demand self-service: Customer should be able to provision the resources as needed without the need for human interaction with the cloud providers.

2. Broad-network access: Services are available through conventional network access to promote the usage by all types of clients such as, mobile, laptop and so on.

3. Resource pooling: Provider resources must be pooled to be assignable and designable to the customers on-demand using a multi-tenant solution.

4. Rapid elasticity: Services can be rapidly and elastically provisioned, to fit the need for services scaling up and down on demand, and even automatically for the client. This characteristic enables a faster response time for the clients, along-side with more cost efficiency to reduce the need for over provisioning resources to handle their different traffic.

5. Measured service: Cloud systems automatically optimize resource usage by analysing the metrics capabilities at a level of abstraction appropriate for a resource. For example, memory usage, CPU usage.

The above-mentioned characteristics of cloud computing are what separate cloud from more traditional infrastructure methods, and thus can help us easier understand the benefits in using it.

2.3 Service models

Cloud services models are divided into 3 models, SaaS (Software as a Service), PaaS (Platform as a Service), IaaS (Infrastructure as a Service). The following section presents more detailed description of each model.

1. Software as a Service

SaaS is a cloud offering in which user can access the vendors software on demand without the need to install the software on their own machine, and the vendor will be responsible for keeping the application up to date. In this type of service users can get the benefit of not having to maintain extra software therefore having more time on developing and maintaining their other services. These services are usually available through an API or using web, there are sometimes options for installing these services on the local machines but requires internet connectivity for making sure user are subscribed to the service.

2. Platform as a Service

PaaS is similar to SaaS in a way that cloud provider gives user access to resources, but instead of giving a software, users are provided with a platform where they can develop, test and deploy their own software. These services reduce the users need for managing their own OS, keeping the system up to dated. Some advantages to these offerings are: Simple cost-effective development of software and applications, high availability, provider managed security and backups as well as less management overhead.

3. Infrastructure as a Service

In this type of services, vendor provides users with access to computing resources such as servers, storage and networking. Users can use their own platforms and applications within providers infrastructure. Some advantages in using these services are: Customer do not have to buy their own hardware, system is scalable, less failure points in the hardware.

To better demonstrate these service models, the following figure shows shared responsibility model between cloud provider, and users.

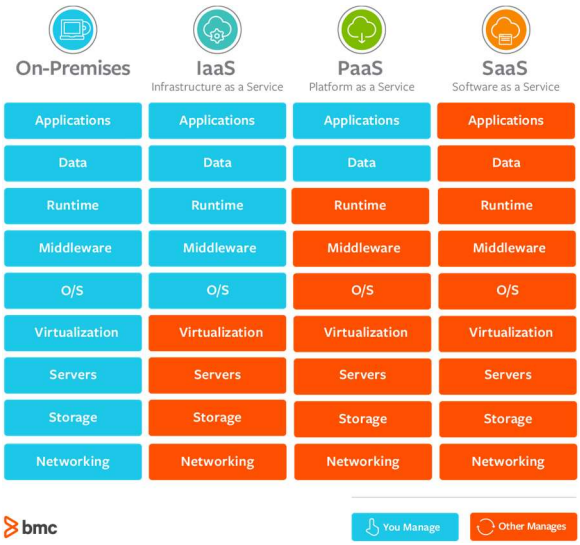

Figure 1. shared responsibility model (Stephen & Muhammad, 2019)

2.4 Cloud vulnerabilities

Cloud computing as any other system has its own shortcomings and vulnerabilities. These systems are still exposed to same issues as traditional data centres, with one difference that responsibility of these threats is shared between cloud providers and the customers. Following section is an overview of some of the most common issues in the cloud. All these issues must be addressed and have policies in place so as to minimize the risks of any security breaches in advance.

1. Internet facing APIs might be compromised.

   APIs are one of the ways to provision, manage and orchestrate the cloud solutions. Like with any other APIs if these communications are not secured, which means they can be used by attackers to exploit user's data and resources. In contrast with on-premises data centres, it is easier to leave these APIs locked out of internet.

2. Poor account access management

   One of the most common mishaps of cloud computing is improper access management. Having long lived and unchanged passwords, unused user account, giving higher than required permissions for users are some of the examples of poor account management that can lead to unauthorized users gaining access to the system easier and faster.

3. Compliance and regulatory

   Moving to cloud does not necessarily guaranty compliance with regulations, in fact it may make it harder to comply with them. The ease of access to the cloud services is also a threat to follow who has access to what data, and for fulfilment of regulations such as GDPR, it is important to know this information.

2.5 Deployment models

There are four main types of deployment models of in cloud computing: private cloud, community cloud, public cloud, and hybrid cloud. (Mell & Grance, 2011)

1. Private cloud

A private cloud infrastructure is an environment used by a single organization, or in the other word, the infrastructure is used by a single tenant. These resources can be managed and operated by a third-party organization or the main organization, or a combination of both (Badger, et al., 2012). These resources can be hosted on-premises or in a different location.

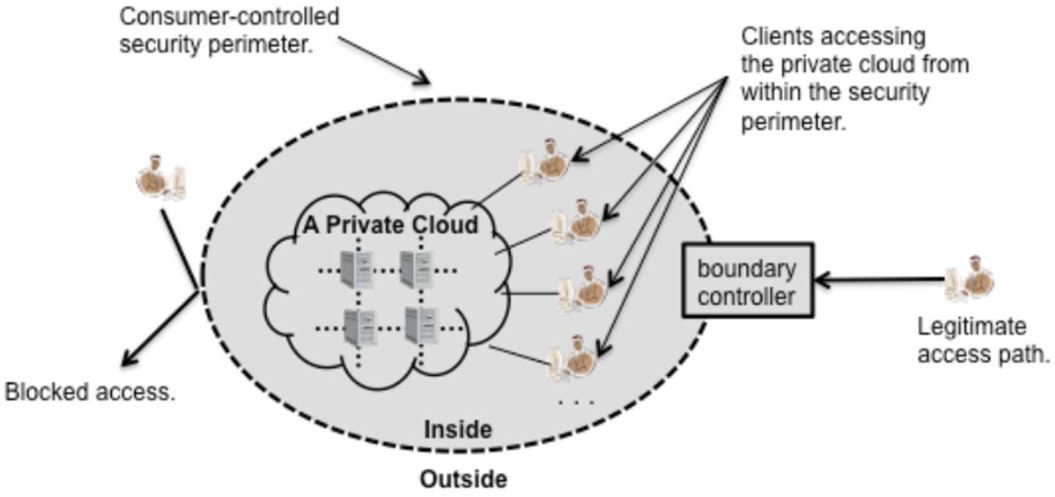

Figure 2. On-site private cloud (Badger, et al., 2012)

As seen in Figure 2 on-site private cloud gives controlled security parameter to the organization consumer. These security parameters have to be defined by consumers, and if it is well defined, it gives more control to the user as of how to control and give access to data. In this model the security of the system as whole can be better than public and community cloud environments since the access can be defined within a specific network and without public access to the network, alongside with reducing risks from multi-tenancy model.

This model of deployment introduces some challenges for the consumer is recommended to be put into consideration: First, this model introduces network dependency for example to one physical location, or cloud network. Accessing these networks from remote networks will require having protocols such as VPNs which can complicate and introduce more risks to the network. Second, in addition to the cloud IT skills, the consumer will need more traditional IT skills to manage devices that are connected to the private cloud. Thirdly, on-site cloud will require a high cost for hardware, and in some cases software setup. Lastly, this type of setup has limitations for the resources, since the storage and compute capacity are based on what has been the prediction of the usage for a certain time period.

2. Community cloud

Community cloud is an infrastructure made for exclusive users who share the same concerns and parameters, such as security, policy and-so-on. The infrastructure may be owned and managed by members of the community and/or third-party vendors. In the context of this section an overview of on-site community cloud is analysed.

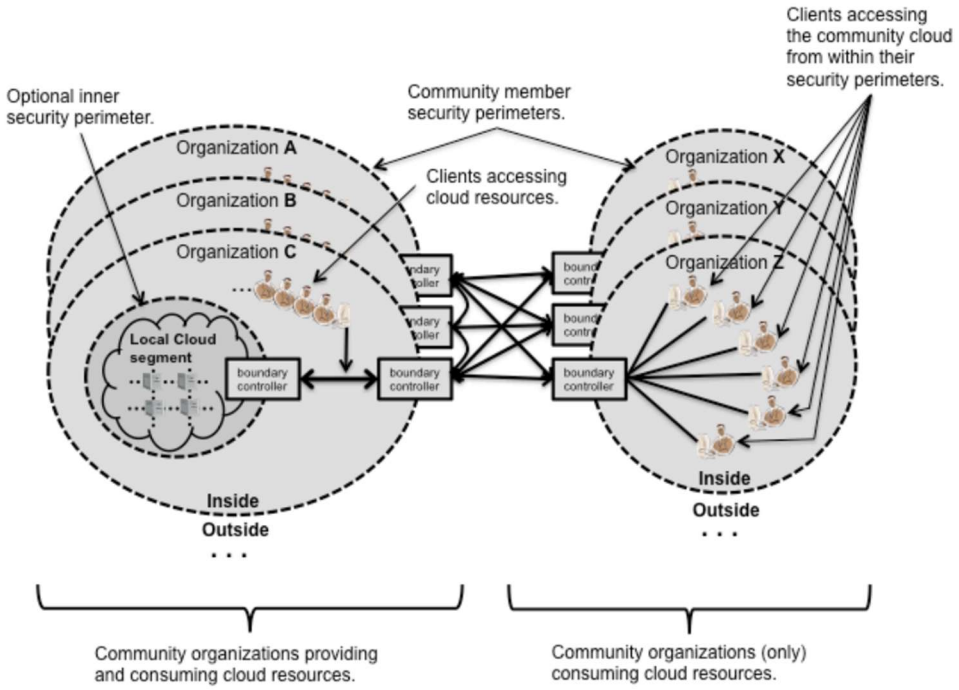

Figure 3. On-site community cloud (Badger, et al., 2012)

As seen in figure 3, it is evident that policies for a community cloud is complicated. For policies and changes all or most members of the community must agree. Although there are several standard specifications available for putting these policies in place. (Badger, et al., 2012)

In this type of deployment as with private cloud same issues persists that should be solved and put into account while implementing the system. In addition to the issues mentioned in private cloud, some of the issues from public cloud are added to the list, such as, risks from multi-tenancy, which will be explained in more detail in the public cloud section.

3. Public cloud

The infrastructure in this deployment is provisioned for public use. It might be managed, and maintained by a business, governments, educational academics. Data centres for reside in the facility of the cloud provider. (Badger, et al., 2012) These public clouds are platforms that use standard cloud computing resources available to remote users, such as virtualization, storage, and others. Some of the biggest companies who offer these services are: AWS (Amazon Web Services), Azure (Microsoft), GCP (Google Cloud Platform), Alibaba Cloud, IBM.

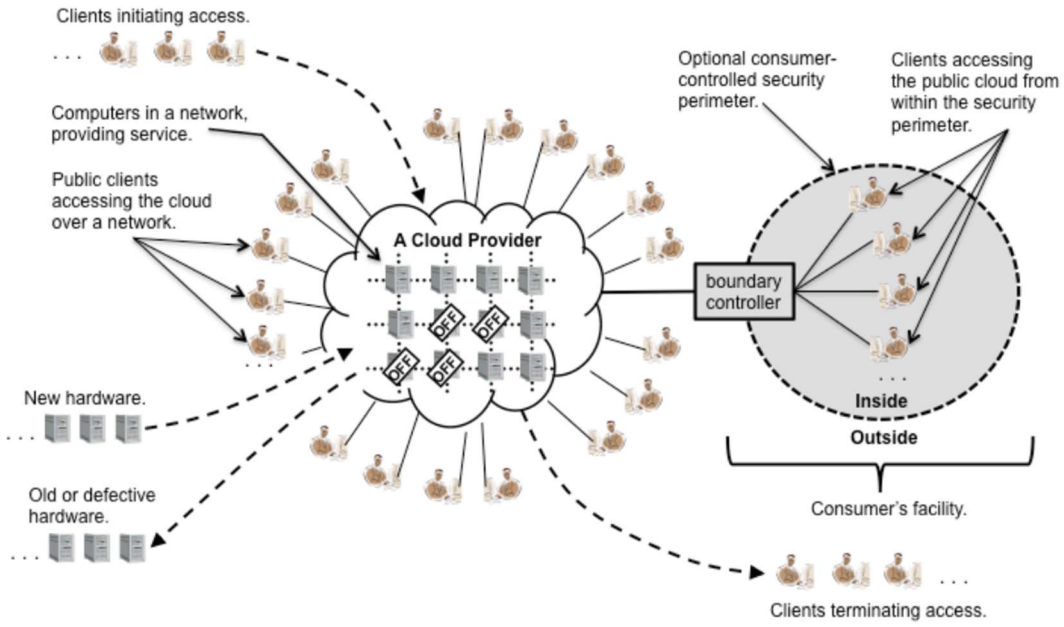

Figure 4. Public cloud (Badger, et al., 2012)

By looking at figure 4, public cloud can be easier understood. In the centre is the provider who creates, maintains the infrastructure and software required for the cloud. On the sides, clients initialize the resources they need for their use case, if required they add more security parameters, such as firewalls. The communication between users and providers are over a network, usually using HTTP and APIs.

These clouds offer many benefits to users and organizations. Some of the benefits are as follows:

1. Ease of adaptation and access to new technologies

By removing the barrier for the need of having many professionals to create private clouds from scratch, companies can get started with cloud computing much faster and efficiently. At the same time providers keep up to date with the latest technologies, making it easier for users to try and access them faster than ever before, and with lower price than making their own.

2. Scalability and flexibility

Due to rapid expansion of resources by public cloud resources, users have unlimited access to as much required capacity as they want. At the same time with possibility to reduce their response times, and down times by deploying their applications in different regions and in case of AWS different AZs (availability zones). Managing high volumes of data and traffic might be challenging in a classic datacentre while in public cloud, managing these are much simpler. For instance, with AWS EFS one can provision 52,673,613,135,872 bytes (47.9 TiB) of storage, while paying only for the storage user uses.

3. Analytics

User can utilize helpful analysis of their resource usage, to best optimize the need for their resources. For example, by looking at the memory usage and CPU usage of a server or set of servers, number of required servers at a time of the week or month can be deduced, to reduce the uptime cost of extra servers at times where usage is less or vice versa.

3 Initial state of the software before migration

This section provides an analysis of the software before migration to the cloud, to identify the shortcomings and the needs for designing a fitted architecture, and security points, while keeping in mind that the system has to be agile enough for future expansion and development. At the same time speed of which the system is developed and getting used in the production is of utmost importance for the project to be able to be a competitive solution, thus over-complication of the system is not an option in the case of this project.

3.1 System structure

Before migration to the cloud, the system was developed as a POC (Proof Of concept). Therefore, in the initial architecture there was a few shortcomings. First, was the issue with scalability of the servers. the first iteration of deploying the application, it was deployed using combination of on-site and off-site servers.

Due to the high cost of servers running using GPUs on the cloud or other hosting solutions GPU servers were hosted on-site with difficulty to scale up on high demand but as a POC setup number of servers would suffice therefore architecture has to account for the cost of these type of servers. At the same time other servers did not have the ability to scale up on demand or based on traffic nor sharing storage between servers, an issue which should be tackled in the migration to the cloud.

3.2 Security analysis

In the scope of POC, some security measurements including encryption at rest and transit had been added, since the project was developed with security first in mind, but in order for the finale product to be in as secure as possible to comply with standard and customer needs more security feature are required.

One of the most important features that needs to be added is stronger data encryption suites for encrypting data at rest on storage and database level alongside with the exist-

ing HDD encryptions, with added centralized KMS (Key Management System) for managing the keys (More information on Chapter 5. Secondly, encryption ciphers for data encryption a transit needs additional research and finding stronger suites for extra security of data. Finally, improving the network security by use of VPNs, removing in-site servers and moving all to the cloud and tightened VPC security.

3.3 Other considerations

One of the benefits to moving to cloud is opportunities for automation in different at the same time with above mentioned there are other issues with the project that needs to be either improved or be added to the project.

One of the areas in which cloud computing excels is opportunities for automation in DevOps and Cloud orchestration, to ease creating new environment and up-keeping and monitoring existing environment in the cloud. Environments in the context of this thesis are, deployment and running applications using set of infrastructures and application provided by cloud providers such as Azure, and AWS. Some of the tools which are required for the application to ease the usage for developers and DevOps team are, CI/CD pipelines for deployment and testing the applications easier in addition to added security due to less human errors and less need for accessing the servers manually for configuration.

Secondly, organising building a cloud service, creating resources and updating using tools such as CloudFormation from AWS or Google Cloud Deployment Manager are important steps. Third, a centralized logging system is required for collecting necessary metrics and application and security logs to reduce management overhead.

4 Choosing the tools and technologies for the project

In this section, information on three cloud provider giants, GCP, Azure, and AWS can be found. Each section is divided into three parts: a short summary of the provider, some services from each one of them and finally a list of pros and cons for the providers gathered from experts and other places in the industry.

4.1 Google Cloud Platform (GCP)

GCP is a cloud provider owned and managed by Google providing with a variety of solution for hosting and deploying applications. (Google, n.d.) GCP is available in 24 regions and 73 availability zones.

Some of the services provided by GCP are as follows:

- App Engine to create and host applications within cloud.
- Support for most common programming languages
- Data analysis tools
- Emphasis on big data and AI, and ML
- CloudKMS (Cloud Key Management System) for encryption
- High number of regions and availability zones

As with any services GCP has a set of pros and cons which are listed below.

Pros:

- Great contribution and reputation in open-source community
- Flexible and lower pricing than other providers
- Well stablished brand in cloud computing
- Vast number of services
- Emphasis of services in AI and ML (Important for the project in question)
- High level of security and compliance
- High number of regions and availability zones

Cons:

- Confusing UI/UX
- High hours of downtime (361 hours of downtime from 2018 – May 2020 (Linke, n.d.))
- Fewer services and products compared to AWS

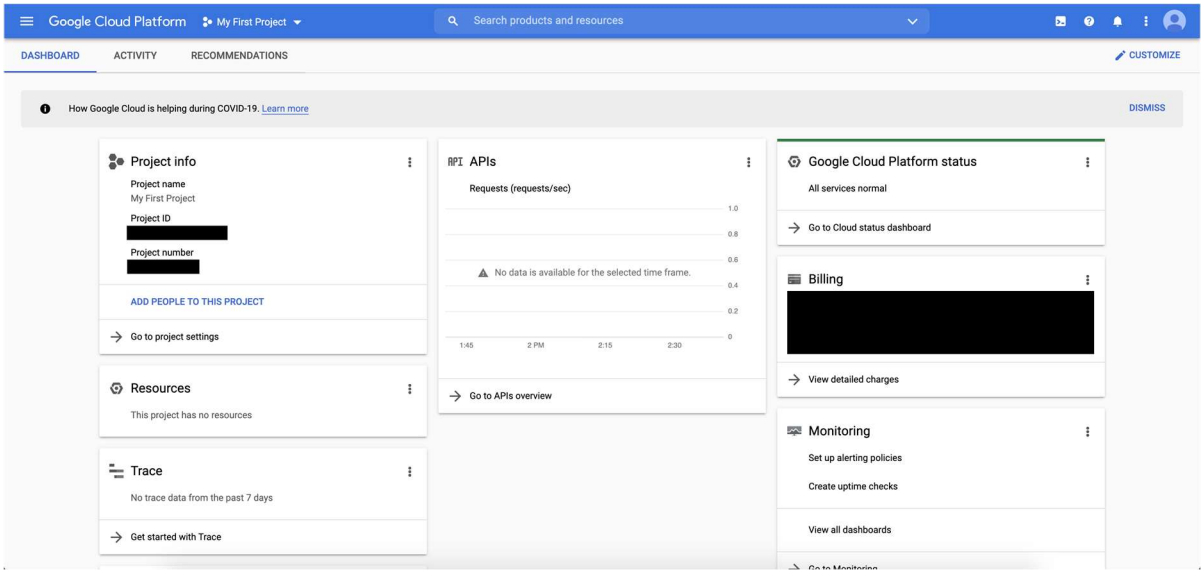

Figure 5. A view of GCP console

4.2 Microsoft Azure

Azure is a cloud provider owned and maintained by Microsoft, and same with GCP they offer variety of solutions for hosting and deploying cloud systems and application. Azure is available in 12 regions which support availability zones, and 60 regions in general. (Micorsoft Azure, 2020)

Some of the services offered by Azure can be found bellow:

- Integrations with Windows Server and Linux
- Machine learning and AI.
- Blockchains services
- Compute
- Key Vault for storing and managing encryption keys.

Pros:

- Offers good and well categorized documentation.
- Good integrations for hybrid clouds
- High number of regions for reduced latency
- Best supports Windows Server.
- High level of security and compliance

Cons:

- Extremely high hours of downtime (Over 1934 hours of downtime from 2018 – May 2020) (Linke, n.d.)
- Offers a smaller number of services in comparison with AWS.
- Best supports Windows Server.

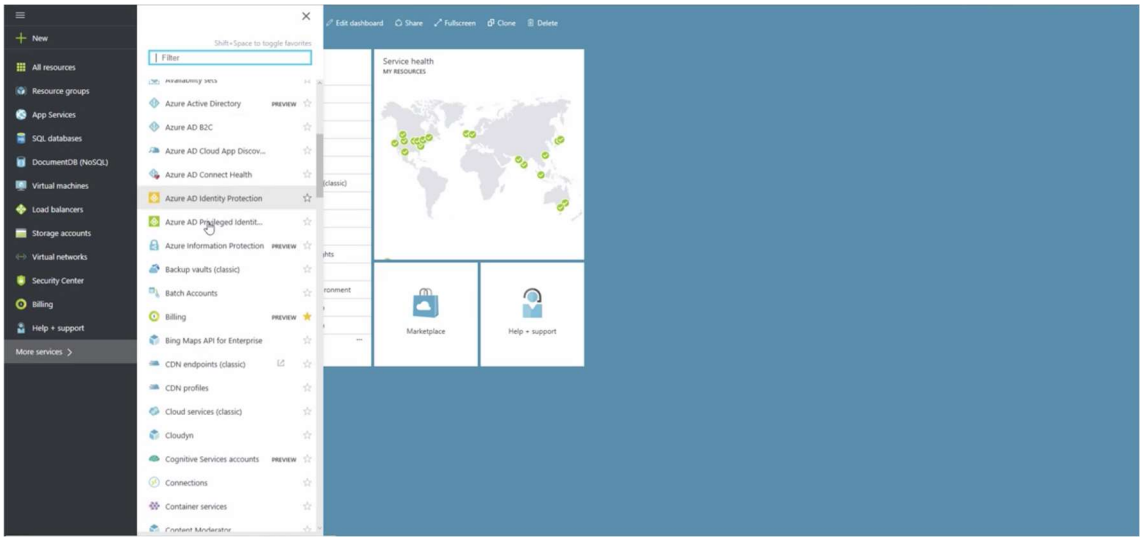

Figure 6. A view of Azure portal

4.3 Amazon Web Services (AWS)

AWS is another cloud provider owned and managed by Amazon started offering IT infrastructure to businesses in the form of cloud solutions, one of the first vendors to offer these solutions to public. AWS offers 23 regions with 69 different availability zones; most of the regions in AWS offer at least three availability zones with the exception of 2 regions. (Amazon Web Services, 2020)

Some of the services offered by AWS:

- Application hosting and CDN
- Compute Resources
- AI and ML services
- Backup and Storage
- Enterprise IT
- High security

Pros:

- High number of available services
- High security and compliancy
- US GovCloud
- Support from most software vendors in AWS marketplace
- Lowest rate of outage between the vendors discussed in this paper (338 hours 2018 – May 2020) (Linke, n.d.)
- Great customer service and on-boarding guidance
- Community support

Cons:

- Higher pricing in compared to GCP and Azure.
- Steep learning curve at the beginning of using services, due to number of offerings.

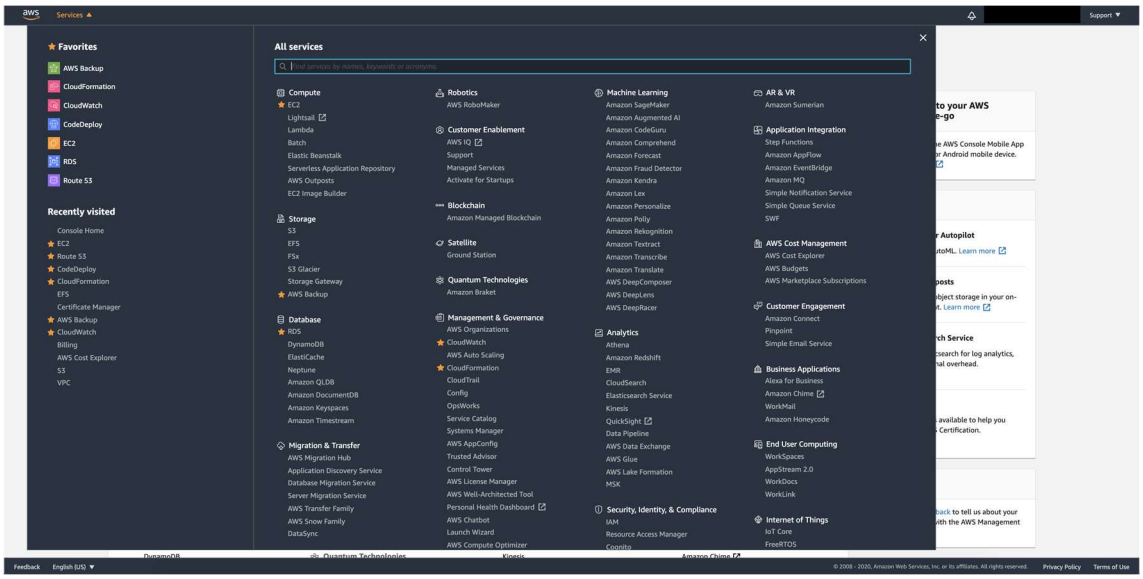

Figure 7. A view of AWS console and services

4.4 Why AWS as main vendor

So far in this section, different cloud vendors have been overviewed one by one and to reduce the overhead of the decision options they were narrowed down to three of the best vendors with most services in their catalogues. All these vendors have their strong suites and are backed by some of the biggest software companies in the world, thus making a decision as of which service to choose from is a difficult question.

In order to ease this decision, first and foremost a list of a pros and cons for each vendor was created which can be found in the sub-chapters above. By looking at those AWS is a clear winner but, security and reliability of these services, and acceptance of these by community and potential customers is of utmost importance. Based on research and interviews with experts in the field of security, AWS have had the best feedback.

Despite the fact that AWS is the vendor of the choice, some issues must be solved before proceeding with the service. One of the downsides to start using AWS is the high pricing specially for things such as, data-transfer out of the cloud costs, pricing for GPU based instances for deploying the AI models etc. Another issue is the learning curve involved in getting started with the service plus finding/choosing the best possible services to be used in the project. So as to solve these issues measures have been taken to reduce the price overhead and optimizing the application and the architecture for the cloud deployment but unfortunately there is no way around the learning curve of AWS and required an extensive studying the system to make the best possible decision.

5 Key Management System (KMS)

Key Management System refers to a system used for storing cryptographic keys trough out the lifecycle of the key from creation to replacement and deletion of the keys. Most users use these systems daily even without realizing it, by saving their passwords in browsers such as Google Chrome or Firefox, or on OS applications like Key Ring on MacOS. A similar need also applies for a software to store the passwords for email addresses used by them for communications with the users or storing SQL passwords and encryption keys without committing them in the CVS and compromising security in case someone gains access to the code repositories. In this section two different systems used as KMS for software and other systems will be introduced with some their main features and usages.

5.1 HashiCorp Vault

HC Vault is a system developed by HashiCorp, a company specialized in DevOps and security challenges in infrastructure based in San Francisco (hashicorp, 2020), where users can deploy the system on major cloud providers using a present template for deployment or deploying their own manually and with preferred architecture. Using this system users can securely store and access their passwords, tokens, encryption keys etc. using a UI, CLI, or HTTP APIs.

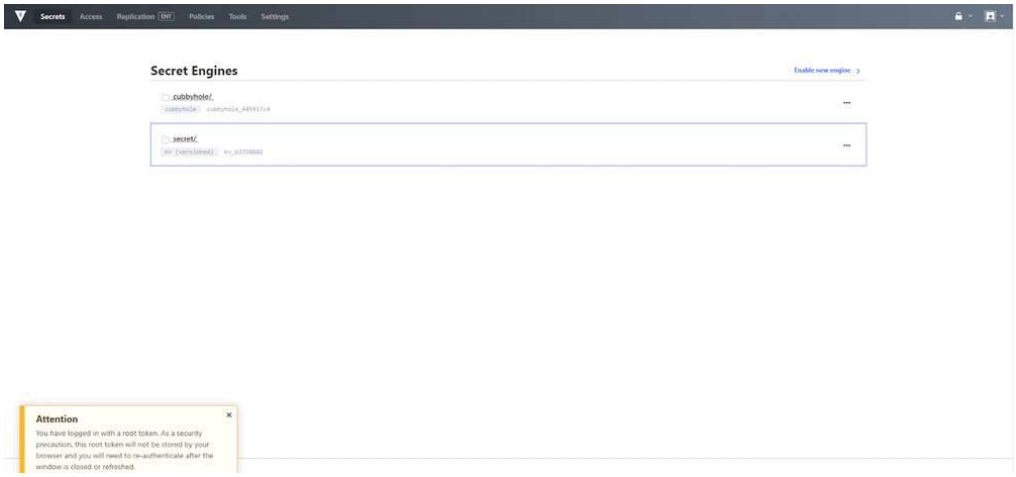

Figure 8. A view of HCV UI

Some of the notable features for using HC Vault which makes it an extremely good system to use compared to other available similar services are as follows:

- Encryption as a Service (EaaS)

One of the main notable features in this KMS is EaaS, also known as transit secret engine in HC Vault, meaning users can easily encrypt their secrets without the hassle of worrying about how the key rotation is done, or writing a library for encrypting their secrets themselves. At the same time, even if the applications are encrypted at rest, this won't stop attackers to not use SQL injection, a vulnerability which allows attacker to influence SQL queries inside and application (Clarke-Salt, 2009), or other types of attacks thus, additional encryption of sensitive data such as credit card info, IP addresses, etc. in the database is necessary to avoid leaking these data to the attacker.

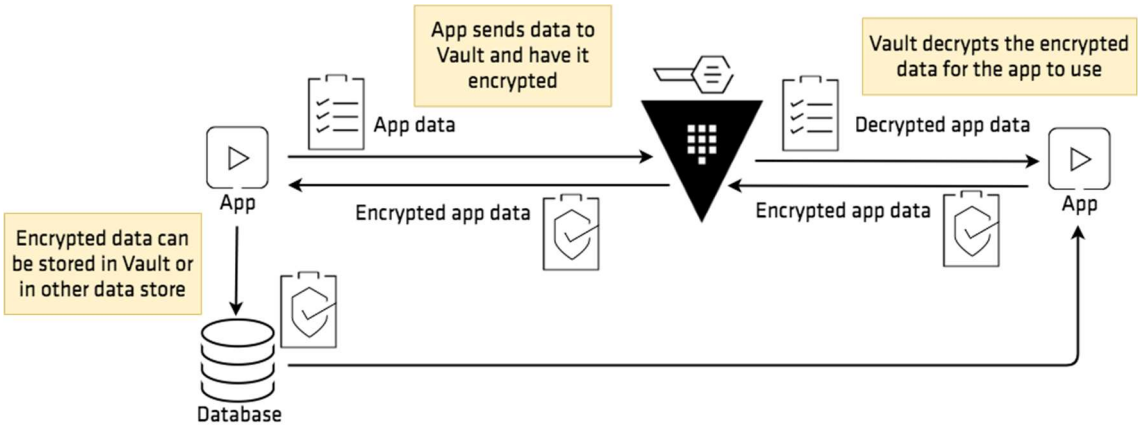

Figure 9. Simple overview of how Transit secret engine works with applications. (HashiCorp, 2020)

As seen in figure 9, to encrypt sensitive data, first application takes the data from users, and instead of directly saving these data to the database directly, application first sends the data to HC vault using either CLI or an HTTP request, HC Vault Transit then encrypts the data using AES-GCM with a 256-bit AES key, or other types of supported keys (HashiCorp, 2020). In case accessing these data is needed, the application before sending the data to the user, it first sends the fields which are encrypted using Transit to the HC Vault for decryption, then gets the decrypted data back from the service and serves it to the user or other required application such as other micro-services. Throughout this

process to avoid vulnerabilities while transiting data from and to HC Vault and application and from application to users and other apps, all the data must be encrypted in transit (not to be confused with Transit secret engine in HC Vault) using SSL and other forms of security in transit which will not be discussed due to security reasons.

- Key Value Secret Engine (KV)

Key Vault secret engine (KV) refers to a database of secrets saved in a format of keys and values similar to DynamoDB from AWS. In HC Vault KV has 2 different ways of saving these data, first way is a not-versioned generic key and one value storage, and second way is versioning enabled for KV storage. (HashiCorp, 2021)

There are benefits and downsides in using either modes of the KV backends, in the first mode of the KV backend since the versioning is disabled there is less overhead of metadata and storage in the backend thus improving the performance of the system and retrieval speed but at the same time in case of any accidental changes, data might get lost, especially if the data used are randomly generated long keys. In the second storage mode the chances of accidental loss of data are marginally less in compared to the first mode subsequent to having versioning enabled and having the ability of retrieving older version, but the performance can be slower than the first version. See figure 10 below:

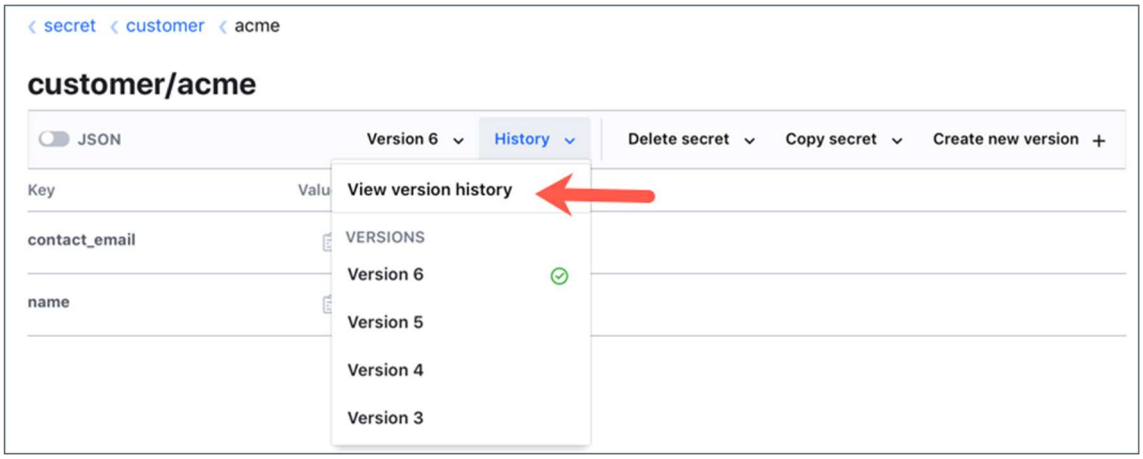

Figure 10. A view from Vault KV storage and retrieving the secret versions (HashiCorp, 2021)

The KV secret engine plays an important role in applications where security is important and there are the same applications running in different environment and they which might need different needs in every environment. First usage of KV is that developers can store their environment variables and their encryption keys in this secret storage and retrieve them when they need it programmatically and with automation without having to hassle with leaving the keys on the servers and unencrypted, at the same time for changing one variable they will not have to login to the servers via SSH, both saving time and increasing overall security of the system. Second usage for this system is the ability to use it as database for important data from users to comply with regulations such as GDPR etc. to make sure everything stored in an effective and secure way.

5.2 AWS Key Management System (KMS)

AWS KMS allows users to create and manage their cryptographic keys and integrate it into wide array of services within AWS and users' custom applications. This service uses hardware security modules, at the same time with accessing to logs in CloudTrail for compliance needs. (Amazon Web Services, 2021)

There are many benefits in using AWS KMS such as easy integration with AWS services, security, and compliance. Integrations with AWS offered services is an important reason as of why KMS was chosen, since using other services such as HC Vault will be highly time consuming if not impossible to use with AWS services such as data at rest encryption for storage units (EFS, S3 etc.). At the same time users can benefit from a highly secured and compliant key management system which has been validated under FIPS 140-2 (Federal Information Processing Standard Publication 140-2) and other compliancy regulations (Amazon Web Services, 2021)

6 Cloud Data Security

In this section, three of the major parts of the cloud infrastructure in the application, Virtual Private Cloud (VPC), servers, and networking, and how they contribute to securing data and making sure the services are well protected.

6.1 Virtual Private Clouds

Virtual Private Cloud in AWS is a service which let users launch their resources in an isolated virtual network. In this service as any network systems such as office space network, all network features are fully configurable, users can assign their own IP ranges route tables, subnets, etc. (Amazon Web Services, 2021) VPCs are the backbone of any applications run on AWS and understanding this service and how to secure it is an important subject.

Using VPCs allows for tighter security configurations on networking. Since users can configure their own networks which are separated from other clients of the cloud services and on a higher level from different microservices of the application. If microservices or set of microservices are defined on different VPCs based on need, on case of any breaches on those services, the whole integrity of the application would not be in danger if networking has been set correctly. On the other hand, if the networking and VPCs are misconfigured, it can lead to dire breaches in data security and data might fall into wrong hands.

At the same time AWS's advanced monitoring systems insures better performance and even stronger security of the overall system. Virtual Private Clouds supports Flow logs where users can receive them in S3 or CloudWatch (Service for collecting logs as a central entity), these give insights and low-level metadata to enable packet level analysis as sources and destination of traffic. Alongside with Flow Logs, Traffic Monitoring service enables users to detect network and security anomalies and get operational understandings. (Amazon Web Services, 2021)

Lastly, an important feature in this offering is Security Groups which act as firewalls. Firewalls are the heart of any secure application deployments. With security groups clients can have full control over inbound and outbound traffic to/from their EC2 (Elastic Compute Cloud). (Amazon Web Services, 2021) For instance, when defining an SSH for servers it's important to have ports used for it as close as possible to an extend which only authorized IP addresses can access these ports which is one of the first things any attacker tries to brute-force and try to gain access to the server.

6.2 Securing Servers

Servers are important subject in securing the applications as it is hosting the codes, in some cases the database resides there, and storing files in most applications. At the same time, it is subjected to a lot of threats and attacks such as DoS, phishing, brute forcing etc.

Before all else to improve on these issues is configuring firewalls properly. Inbound and outbound traffic running through the servers should be limited as much as possible and restricted mainly to the application related ports such as port 80 and 443 for HTTP and HTTPS, respectively. In Amazon Web Services security groups acts as firewall rules with a UI and CLI configurations, alternatively users can use Linux based firewalls such as UFW on ubuntu.

One danger of any system and computer is infestation with malicious files, viruses and malwares etc. which protection against them is crucial for system. There is many anti-virus software in the market ran on all platforms like Ubuntu and Windows both servers and consumer level. After selecting a suitable software based on need users need, like computational power, budget, etc. the software should then be configured to have active and scheduled scan of the servers at least once a day, for an added security scans can be added to the endpoints of the API which handles files.

Lastly, to reduce the dangers of brute-force, DoS and other types of attacks against the servers and services, Intrusion Detection System (IDS) can be introduced to the servers. Intrusion detection system in a simple language is a software which analyses the incoming requests to the servers and network and if it detects any abnormalities or high

amounts of requests from a single source to the parts of the network higher than allowed threshold, it will block the requests coming from said source. This will help to ensure if an attacker is brute-forcing SSH port (22) in order to gain access to the server or just simply a user is trying to bring the system down in a DoS attack, they are either stopped or at the bare minimum they are slowed down as much as possible. One of the more popular and open-source examples of these software is Fail2Ban made for Unix and Linux systems where system admins can configure them in the way that fits their applications the best.

6.3 Encryption

Encryption refers to changing data from plaintext and readable formats in to ciphers texts for added security. Encryptions ciphers have been in development and improvement throughout centuries with early versions just being a number deviation from the characters in the language which were not too hard to crack the codes. Today the encryptions have gotten much more sophisticated to a degree were decrypting some these ciphers even with super computers will take hundreds of years to guess the keys for deciphering the messages.

As a result, In the current age where data has become more important and more available than before, encryptions are getting more and more attention in order to secure the data, which is being saved, from sensitive governmental documents to data about users' personal information, photos, etc.

In the scope of this thesis two types of encryption of data will be discussed which are more widespread used to safeguard data in software and applications, which are encryption of data at rest and in transit.

1. Encryption of data at rest

As the name suggests data at rest refers to the state of data where it is stored on the storage system while it is not being used. At this stage of storage if an unauthorized entity gets access to the data anything which is stored as plaintext would be easily readable. To better visualize this, let us assume an external hard drive laying around in the

office, if anyone picks it up, they can easily plug it into their own computer and read its content as private as it might be.

Now to fix this issue, one can introduce a full encryption to the hard drive, so if someone were to plug it into their personal computer, they would see scrambled bits of data rendering it useless to them until the owner would enter the password to decrypt the data and make something meaningful again. At the same time users should bear in mind that what they choose as the method as the cipher is of utmost important and not to choose a previously broken cipher and the length of the key to decipher the data.

In order to secure our data on most cloud providers offer services to achieve this in easy way. On Amazon Web Services customers can create a new key at KMS system and while creating their storage in any of the services most notably S3, EFS and EBS enable encryption option using the key they have created. If a key is not generated it will default to a master key auto generated for that service. See figure 11 below:

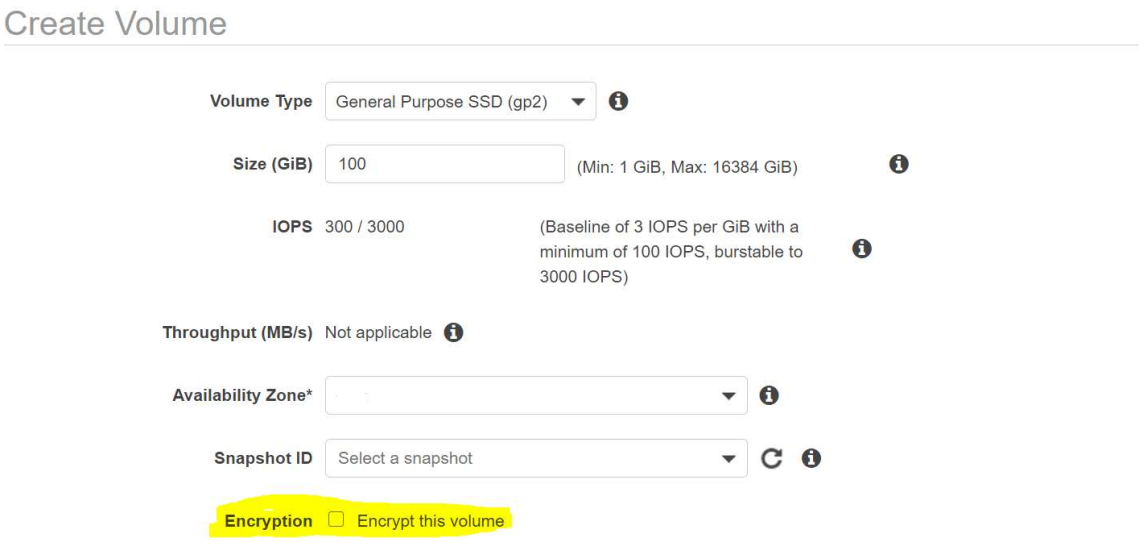

Figure 11. An example of how to activate storage encryption on EBS.

With having the storage secured, still there is one problem to deal with while data is at rest, which is when the storage is in use someone unauthorized might still be access the data. To visualize this, assume the following scenario: an employee is working on computer with some sensitive documents stored on it and suddenly their phone starts to ring,

at which point he/she will leave the laptop unattended while taking that call. In this scenario even if the hard drive is encrypted at rest, since it is in use all the data is in plaintext format, so anyone passing by can view or copy or modify the sensitive documents.

As to solve this issue, another layer of encryption should be introduced to the system. In this layer information will be classified based on importance, how often will they be accessed, are they important to indexing of the data and so on. Then they will be assigned different types of ciphers then they will be encrypted based on their classification. Finally, our application can encrypt and decrypt the data whenever it needs to use them, otherwise it will stay as unreadable mix of bytes.

2. Encryption of data in transit

Another vulnerability where data could be exposed is when it is being transferred from one place to another using network protocols. As a real-world example, one can imagine this as a wire for electricity, if the wire is not coated and protected in case anyone touch it, they will be electrocuted.

To avoid getting electrocuted as in the example data which are being moved must be encrypted for transit. In order to achieve that two main methods are used first and older version known as SSL (Secure Sockets Layer) and its successor TLS (Transport Layer Security).

To understand better how SSL/TLS connections work, lets stablish some ground works on how encryption works, what are symmetric and asymmetric keys, and benefit and usages for each. Encryption in general is divided into two groups.

The first group uses a symmetric key, where the message is encoded using ciphers with key A and in order to decode the message the same key (A) must be used for deciphering. An example of ciphers is AES (Advanced Encryption Standard) using keys of sized 128, 192, 256 bits to encrypt and decrypt data in 128 bits. (NIST, 2001) A good example of encrypting and decrypting data is when someone sets a password for his or her files in an archive, another person must have the same exact key to unzip the file and view the content.

The second group uses asymmetric keys to encode the message. In this group data is encoded using a key commonly referred to as public key and it is then decoded using another key known as private key. In this scenario, a person only needs to receive messages, such as a public form, from multiple entities. The person needs to keep these messages unreadable for each sender although they can all be decrypted at once. To achieve this, the person could send a new symmetric key to every unit and ask them to encrypt the messages using that key. However, the person would run into a problem of having to memorise or keep track of many keys at the same time.

To fix this problem the user can create a pair of private and public keys and only send the public one to another individual who needs to fill in the form. These individual can then, in turn, encode their messages and send them back. Here only the person who has access to the private key can decode the messages and use them and the public key is only used for encoding the messages.

In the example above, even though the key returns in a safe manner to the person who sent the original one, users have no way of making sure who this original sender is. In other words, it is not possible to make sure that someone else is not trying to steal your messages using his or her own key. To ensure the integrity of the system is intact, a third party called Certificate Authority (CA) will be added. CA signs the keys and verifies the identity of the original message sender. A real-life example could be notaries validating documents.

There are many benefits and downsides to using both encryption methods. Using symmetric keys user will have better performance in terms of speed while the other type relies on heavier calculations for decoding messages. On the other hand, user can benefit from a better security with the asymmetric keys, since it does not rely on the shared key being communicated and risk getting exploited at this stage.

For securing the data in transit, SSL/TLS uses bests of both worlds so as to ensure security, integrity, and performance of the system. To start a connection a handshake is exchanged between client and server, in which using the public key certificate and the available cipher specification on the client they secretly agree on a shared key to be used for the continuation of their communication.

Another consideration in using Transport Layer Security is the cipher specs available to the client. Some of the older browsers such as Internet Explorer or older versions of modern web browser might be using outdated and less secure ciphers, which might cause vulnerabilities to the overall systems. Tackling this, I have made a basic assessment system for the project while designing the architecture. The system is by answering few questions as follows:

1. What is the role of the microservice? (Public home page? Authentication system? Admin Panel? API?)

2. What kind of data is going to be communicated? (Basic user data? Classified information? Basic information about company to everyone?)

3. How important is accessibility and browser coverage for the microservice?

4. How impacted are the data based on GDPR?

5. Is the microservice going to be internet facing or internal only?

After answering these questions, the microservice is then classified in the proper grouping and can easily decide which ciphers to allow and which ciphers to ban on the server. The more sensitive the microservice data, only more secure ciphers and will have less browser coverage. For configuring these ciphers, one can do it directly on their web server like Nginx or Apache or benefit from easy tools available on the cloud providers like AWS's ready-made security policies found in load balancers.

7 Client Data Isolation and Segregation

Privacy is a backbone of any application, and developers must safeguard client data and take measurements to keep them safe. In this chapter, analysis of how to properly keep different client's data separated from each other in a cloud environment and application level will be discussed. These needs and implementations might differ from application to application based on level sensitivity of which is defined by either client and or developers.

To isolate the data there are three primary models often used, silos, hybrid and pool model. These segregation models rely on both cloud level, and application and database level knowledge and implementation to be achieved.

Disclaimer: while writing this chapter SaaS Storage Strategies (Amazon Web Services, 2016) white paper from AWS has been used as the source material for getting data segregation ideas in the cloud.

7.1 Silo model

Silo model refers to an architecture in which data are completely separated for each client. To make this simple, one can imagine an office space in which every company has their own office with walls and keys to enter them and no other company can enter their office. For implementation of this model developers and system admins can go about it in different ways.

First way of achieving silos which require least to no dependency to cloud is to separate the databases for the clients. By separating each client databases from each other, a better guarantee is given as if one client table or database is for any reason under jeopardy the rest of the databases would remain safe. To be more accurate, one of the most important things to be achieved by going to silo mode is the reason mentioned above. Despite the fact that separating the databases does not rely on cloud solutions and can be done anywhere, it does add an enormous overhead to the code and maintenance of it.

Second solution to silos is by adding a new Virtual Private Cloud (VPC) for every client with their own set of resources such as database, servers, load balancers etc. as seen in figure 12. This architecture enables a lot of possibilities for the application like developers can go on developing the system normally and the pass the overhead of managing silos to DevOps team. DevOps team can then create these VPCs using CloudFormation or other orchestration tools.

Additional benefit of using this architecture is that all clients can still inherit and share the resources, which needs to be centrally managed in the master VPC. To mention one of these shared resources, is centralized logging system, to enable getting all the security, application and other logs and metrics in one place, instead of having to go to every VPC separately. See figure 12 below for more information:

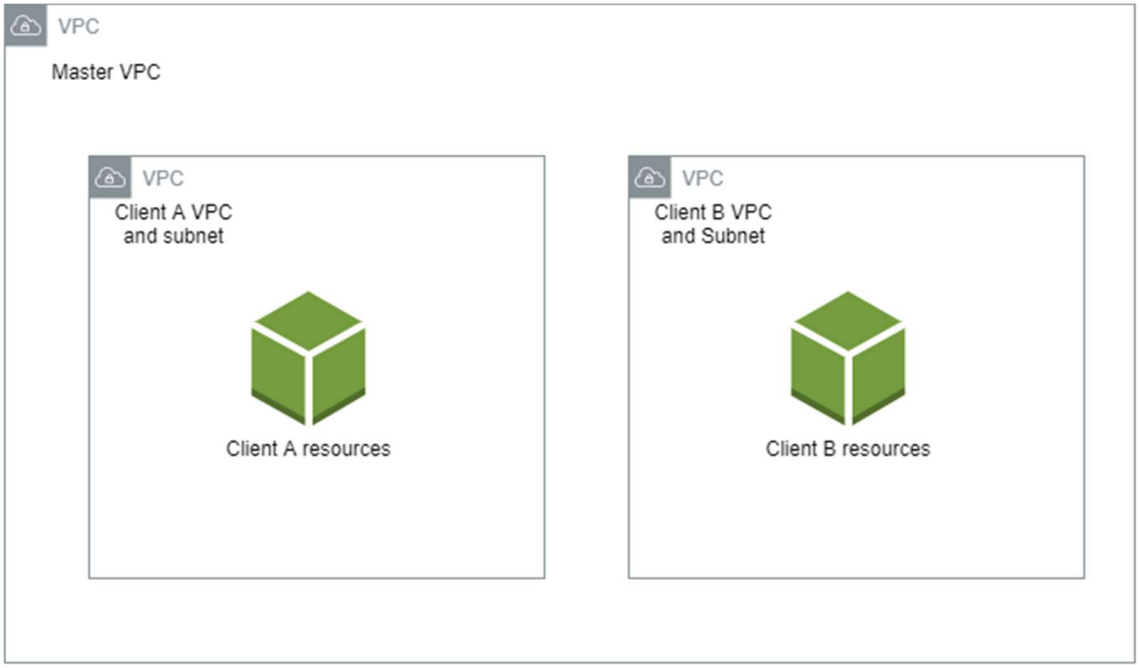

Figure 12. A visualization of Silo model with creating a separate VPC for each client.

Last isolation solution discussed in this chapter is by separating the resources completely with use of organizations. Segregation using organizations on the cloud is similar to separating the VPCs with the exception that DevOps teams can manage billing easier and offer more security than other methods discussed.

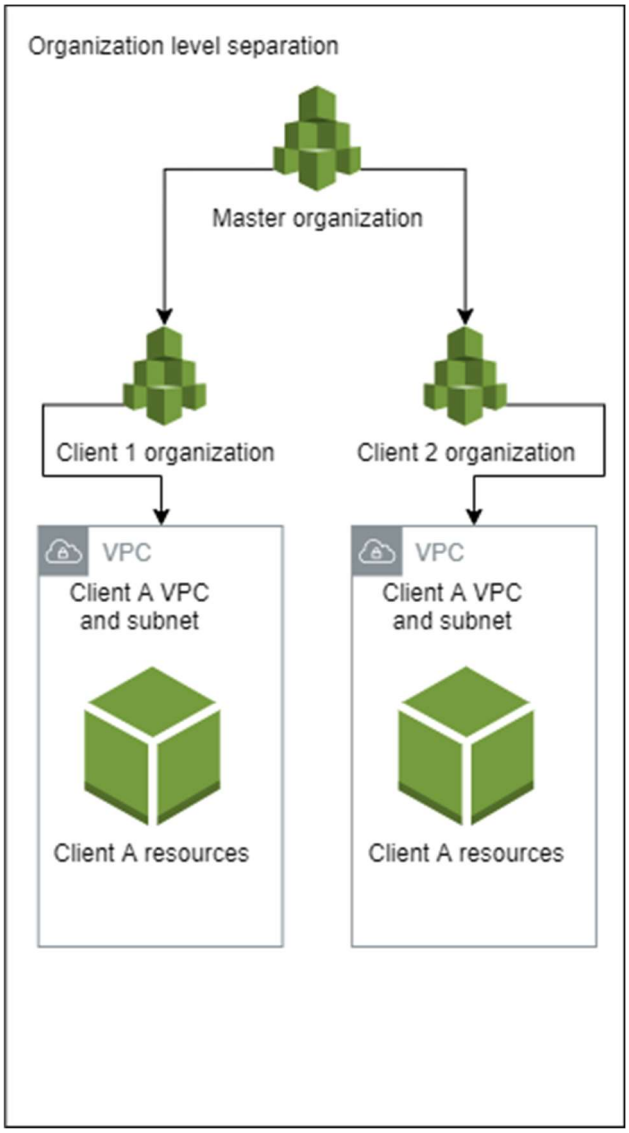

Figure 13. A visualization of Silo model with creating separate organizations for each client.

7.2 Pool and hybrid model

Pool model in data segregation is an architecture in which different client's data are all in the same shared resources. For this architecture to work there must be some measurements to ensure client A cannot access client B's data and so on. In the office example in chapter 7.1, a shared office space where multiple companies are working, no one can access another worker's personal belongings in the lockers and on their computers although all of them are sharing the same office space.

There are many drawbacks and benefits in using pools as opposed to silos. Firstly, pool solution is more cost effective than silo both in hosting and development costs. On the other hand, silos offer more security than pools, since if one silo is malfunctioned or under attack, other silos are safe, but in pools all might possibly suffer in case anything goes wrong.

For a multi-tenant solution to work, a secure Role Based Access Control (RBAC) is critical. This type of access control as the name suggests helps to make sure users can access only their own resources and within the privileges assign to them.

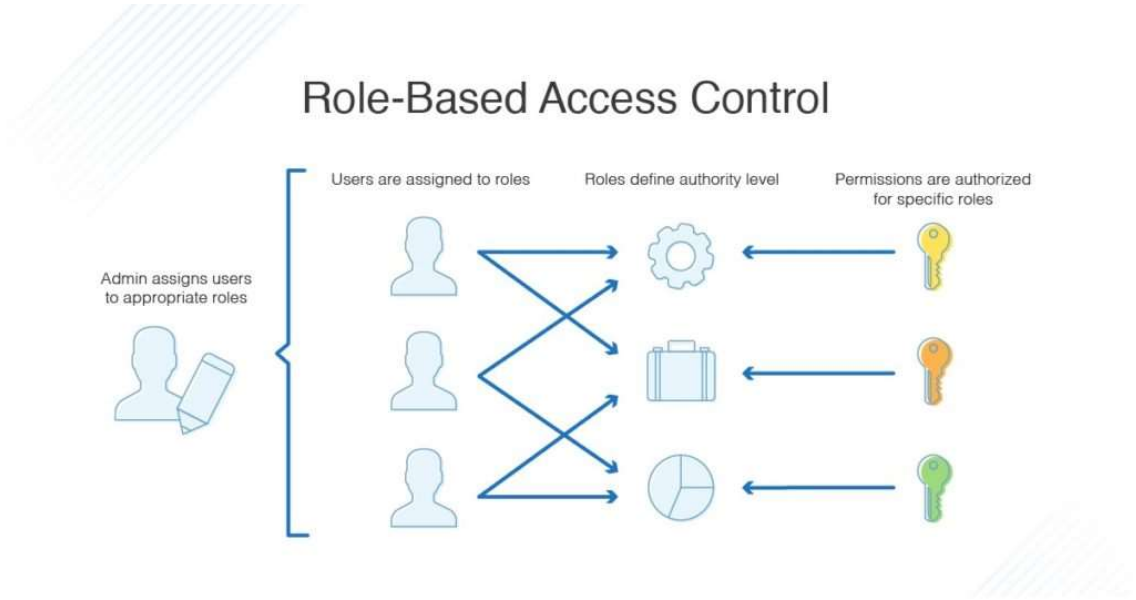

Figure 14. RBAC design diagram (dnsstuff, 2019)

As seen in figure 14 in RBAC an administrator first creates roles alongside with permissions for the roles as of what they can access, what operations (read, write, etc.) the roles are permitted to do. The roles are then assigned to users and based on that users can access the resources on the application.

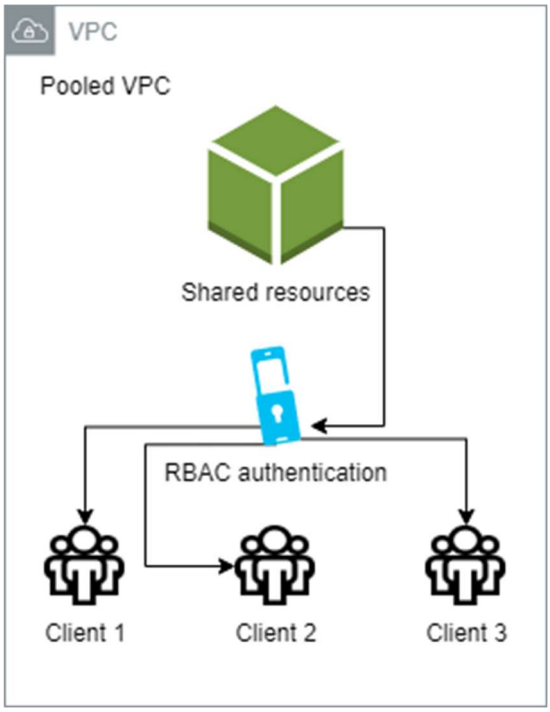

Figure 15. A sample of pool design diagram

With having pools with RBAC enabled and a combination of silo model a door opens to a new model known as hybrid model. For better understanding this model let us get back to the office example. In this scenario companies A and B do not mind sharing the same office space with each other the same with companies C and D but group of the two companies want to have complete separation and privacy from each other. As a result, each group of the companies gets two separate office spaces to share.

The benefits of using this model are that at the same time with providing a very good level of security, it can be more economical than having a silo per each client and increase the types of offerings for the business teams to better sale the system if it is the goal.

8 Project results

For evaluating success of the project there has been many things to be tested due to the complexity of the project. In this chapter some of the tests devised with their results are analyzed to evaluate design choices and their implementations throughout the thesis. These results are presented in two ways: first, system tests and second, reviews from other stakeholders and project owner.

8.1 System tests

An important thing to be tested in a cloud environment is system scalability. In order to assess this, a simulator with multiple nodes was created would generate different types of traffic at random and scheduled. Using this simulator, we know how well a single server can handle a request under stress, and if the scaling groups acted correctly in these circumstances. After a few iterations of optimizing the servers and bug fixes applied, speed of up scaling and overall reliability of system was increased to have the best system possible.

In addition to the test above, system recovery must also be tested. For this, different parts of the system were shut down or deleted in random (not in production systems), to ensure all would recover successfully and without too much downtime.

Next set of tests are specific to the codes being delivered, for this, we used CI/CD pipelines to for automating this task. These pipelines can help users to run automated tests before running the automatic deployment and if the tests would not be successful, the system can report to the developers to fix the issues.

The final test set discussed in scope of this thesis is testing how the system operates in the long run. The concept of this test is simple, just letting it run for a few months while users and simulators use it normally. The main issue is if anything goes wrong, the next tests will take a few more months. Therefore, the previous tests mentioned should have run successfully.

8.2 Reviews from stakeholders

This section presents the reviews of the project owner and the head developer. They were evaluating this project since the beginning.

Hemmo Latvala (Head Developer of the project):

”Amir's done an excellent job studying and adopting best practices for delivering, maintaining and securing cloud-based services. His work has allowed our company to scale from being hosted on a couple of computers at the office to being able to securely provide our service anywhere in the world with no down-time or limitations.”

Oskari Heikel (Chief Operating officer and Co-founder):

”Year and half ago we choose AWS as our SaaS solution platform provider. Before the migration, we carefully evaluated the tools and technologies for the project. Amir was integral part of evaluation team and did the comparison between different cloud service providers and finally took part to the final decision between different service providers (Google Cloud, Microsoft Azure and Amazon Web Service AWS). After the careful evaluation we made the conclusion that AWS delivers the best tools and technologies for our needs.”

“Since we started the migration project Amir has been designing the architecture and the system security in co-operation with our Lead Developer, AI programmers and service development team. In addition, he has been the main backend programmer during the migration. Along this complex process, he has been communicating constantly with the senior engineers at AWS.”

The initial phase of the migration is now ready and in production use. We are still developing the system, optimizing it and improving the reliability and adding new features and services for our customers. So far, the AWS powered SaaS service has been reliable, with practically zero downtime. Thanks to our highly skilled developers the migration project went really well, also the support we got from AWS was vital. They have been helping

us to choose the right tools and technologies during the project and continuing support even though the initial phase of the migration project is ready and in use.”

9 Conclusion

It is not easy for a company to migrate to cloud technology, when there are multiple vendors and each vendor offering many services within their catalogues. The purpose of this study was to carry out research to find out how to prepare for the migration and how to design a secure architecture for a company.

In this final year project, the service models and the vulnerabilities of cloud and deployment models were studied. In addition, the needs, which the company has to fulfill, were discussed. In this study, three major cloud-computing providers are introduced and analyzed by listing their pros and cons.

This study aimed at finding out how the company can secure the data and provide the maximum privacy for the users based on their needs and requirements. Key Management systems and their usage in the applications were found to be the entry point of secure encryption system. Moreover, cloud data security and how to secure the fundamental parts of each application were studied. Finally, how to segregate the data to minimize malicious threads to the application while ensuring maximized privacy for the users and their use cases was studied.

References

Amazon Web Services. (2016). *SaaS Storage Strategies.* Amazon Web Services.

Amazon Web Services. (2020). *Amazon Web Services | Cloud Computing Services*. Read 8 November 2020.
URL:https://aws.amazon.com.

Amazon Web Services. (2021). *Amazon Virtual Private Cloud.* Read 7 January 2021. Amazon Web Services
URL:https://aws.amazon.com/vpc/?vpcblogs.sort.by=item.additionalFields.createdDate&vpc-blogs.sort-order=desc.

Amazon Web Services. (2021). *AWS Key Management Service (KMS).* Read 31 January 2021. Amazon Web Services.
URL: https://aws.amazon.com/kms.

Badger, L., Grance, T., Patt-Corner, R., & Voas, J. (2012). *Cloud Computing Synopsis and Recommendations.* Gaithersburg: National Institute of Standards and Technology.

Clarke-Salt, J. (2009). SQL Injection Attacks and Defense. In J. Clarke-Salt, *SQL Injection Attacks and Defense* (p. 1). Syngress Publishing.

Dnsstuff. (2019). *RBAC vs. ABAC: What's the Difference?* Read 31 October 2019.
URL:dnsstuff:https://www.dnsstuff.com/rbac-vs-abac-access-control#:~:text=The%20primary%20difference%20between%20RBAC,%2C%20environment%2C%20or%20resource%20attributes.&text=ABAC%2C%20RBAC%20controls%20broad%20access,takes%20a%20fine%2Dgrain%20approach.

Google. (n.d.). *Google Cloud Computing Services*. Read 11 May 2020.
URL:https://cloud.google.com/.

Hashicorp. (2020). *About Us*. Read 30 December 2020. from hashicorp.com.
URL:https://www.hashicorp.com/about.

HashiCorp. (2020). *Encryption as a Service: Transit Secrets Engine.* Read 30 December 2020. HashiCorp Learn.
URL: https://learn.hashicorp.com/tutorials/vault/eaas-transit.

HashiCorp. (2020). *Encryption as a Service: Transit Secrets Engine*. Read 31 December 2020. HashiCorp Learn.
URL: https://learn.hashicorp.com/tutorials/vault/eaas-transit.

HashiCorp. (2021). *KV Secrets Engine*. Read 31 January 2021. Vault project.
URL:https://www.vaultproject.io/docs/secrets/kv.

HashiCorp. (2021). *Versioned Key/Value Secrets Engine.* Read 31 January 2021. Learn hashicorp.
URL:https://learn.hashicorp.com/tutorials/vault/versioned-kv.

Linke. (n.d.). *Comparing the cloud giants: Uptime and reliability*. Read 11 May 2020.
URL:https://www.linkeit.com/blog/comparing-the-gigants-of-cloud-uptime-and-reliability.

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing.* National Institute of Standards and Technology.

Micorsoft Azure. (2020). *Cloud Computing Services* | *Microsoft Azure*. Read 8 November 2020.URL: https://azure.microsoft.com.

NIST. (2001). *ADVANCED ENCRYPTION STANDARD (AES).* NIST.

Stephen, W., & Muhammad, R. (2019). *BMC Exchange.* Read 9 June, 2020.
URL:https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose.

Wyld, D. C. (2009). *Moving to the Cloud: An Introduction to Cloud Computing in Government* (1st Edition ed.). Southeastern Louisiana University: IBM center for the Business of Government.