

# Research report

## << *Centralized Firewall* >>

<< *Semester 6 - Infrastructure* >>

<< *Fontys - Eindhoven* >>



### Version

Version	Date	Author(s)	Amendments	Status
1.0		Group 1		In progress
2.0				

## Table of content

1. Sub-question 1:	4
1.1 Introduction	4
1.2 Why are requirements important?	4
1.3 The benefits of high-quality requirements	4
1.4 How did that phase go?	4
1.5 Conclusion	4
2. Sub-question 2:	5
3. Sub-question 3:	6
3.1 Introduction	6
3.2 What services are important to have?	6
3.3 Comparison of service names from AWS to Azure	6
3.4 How did that phase go?	7
3.5 Conclusion	7
4. Sub-question 4:	8
4.1 Introduction	8
4.2 Why would we want to connect a datacenter to the cloud?	8
4.3 Amazon web services	8
4.4 Microsoft Azure	8
4.5 Conclusion	8
5. Sub-question 5:	9
5.1 Introduction	9
5.2 What is Terraform, and for what is it used for?	9
5.3 What is CloudFormation?	9
5.4 For what is Terraform/CloudFormation used?	9
5.5 How can we apply Terraform or CloudFormation in our project?	9
5.6 Conclusion	9
6. Sub-question 6:	10
6.1 Introduction	10
6.2 Why is it important that the technology follows the design?	10
6.3 The benefits of the right Agile methodology	10
6.4 How did that phase go?	10
6.5 Conclusion	10
7. Sub-question 7:	11
7.1 Introduction	11
7.2 When do you use a proof of concept?	11
7.3 Benefits of Using Proof of Concept & steps to write a PoC	11
7.4 Comparison between PoC & Prototype & MVP	12
7.5 How did that process go?	12

7.6	Conclusion.....	12
8.	Sub-question 8:.....	13
8.1	Introduction.....	13
8.2	CI/CD.....	13
8.3	Terraform Cloud.....	13
8.4	Conclusion.....	13
9.	Sub-question 9:.....	14
9.1	Preface .....	14
9.2	Requirements .....	14
9.3	Conclusion.....	14

## 1. Sub-question 1:

What are the requirements?

### 1.1 Introduction

This was the first and very important sub-question, which we needed to work on to have some requirements for the Sogeti's project. This was very crucial because, basically, it would have created the path that we needed to follow throughout this project. We really need to come up with important requirements, which are going to be crucial and we will have to at least some of them, to be more specific the most important ones.

### 1.2 Why are requirements important?

They establish a foundation for product vision, scope, cost, and schedule and they ultimately must target finished product quality and performance. Requirements provide stakeholders such as marketing, electrical engineering, mechanical engineering, and software engineering with one uniform vision and set of goals. Each stakeholder will be able to understand the requirements and hold realistic expectations for the final product. The final product should come with no surprises in terms of design and functionality, ultimately yielding a successful and profitable product launch.

### 1.3 The benefits of high-quality requirements

A product that is developed with a dedicated requirements process will reap many added benefits. Products with well-written requirements will have fewer defects and cleaner designs. Identifying essential requirements from the start of the project will significantly reduce project rework and mitigate technical risks prior to the product's release. As the project successfully passes through each state of the product development lifecycle, the overall cost will be reduced and the product's speed to market will greatly increase.

### 1.4 How did that phase go?

We were able to create a Moscow table, including all the requirements that we needed to include in the project. The whole process took more time than we expected because we really needed to investigate the possible requirements. We had a little help from my coach this semester and mainly from our stakeholders, but we were the main people to create a full list of the requirements. The main points were certainly to have specific requirements, but as well to get a better understanding of every single requirement. Our initial idea was to create a first draft of the requirements where every single member of the group had his opinion. We really made a lot of changes during this phase in requirements due to the fact that we really wanted to have the most accurate requirements for the project. Another thing was to prioritize which requirements were a must and which ones were secondary requirements. In the end, we had around 11 requirements in my list, which we believe is alright. We really believe that this phase was extended more than expected, but in the end, this was the right choice.

### 1.5 Conclusion

The requirements are crucial when starting a project if you want to achieve a desirable result at the finish line. They should include all the features and functions a product should have. Requirements should be comprehensible for all interested parties (development team, stakeholders) and be free of any ambiguities (all the stakeholders should understand requirements in the same way). Proper requirements not only help in understanding and delivering better user needs but also help build the pathway to improved solutions. Requirements in project management also sets the base for budget, schedule, cost, and operations. For

all the stakeholders, end-users and other partners to have a holistic and similar view of the product or solution, it is of utmost importance that proper and meaningful requirements have been captured and analyzed.

## 2. Sub-question 2:

How can Sogeti handle the requests for a centralized firewall Operations?

### 3. Sub-question 3:

What services do we need in AWS and Azure?

#### 3.1 Introduction

This is the third sub-question, which related to the cloud environment. A large selection of services is available in the cloud. The question was which services we generally need to set up an environment with a firewall solution that can be used in a multi-cloud environment. It was known that AWS has the highest priority, so from that perspective we examined which services we need.

#### 3.2 What services are important to have?

Based on the requirements it's important to have at least the following services:

- EC2. This service will be used because we need Windows/Linux based webserver and databases for the environment.
- VPC. We need VPC's to setup the network where the servers and firewall will be active in. The security best practices (ACL's and security groups) can serve as an extra security layer in the network.
- IAM. This service will be used because the access to AWS resources needs to be securely controlled.
- CloudWatch. We need this service to monitor the performance and security of our resources in the environment.
- Cost Explorer. This service is useful because the stakeholder expect that we keep track of the cost and usage in our environment.

For additional security and observation in the cloud, the following services are useful:

- Network Firewall. A network Firewall can be used for setting up a more advanced firewall solution for the multi-cloud environment.
- Firewall Manager. This service can be useful for setting up a more advanced firewall solution for the multi-cloud environment.
- Security Hub. A Security Hub can be used for setting up a more advanced firewall solution for the multi-cloud environment.
- Detective. This service can be useful for setting up a more advanced firewall solution for the multi-cloud environment.

#### 3.3 Comparison of service names from AWS to Azure

Since Azure is also involved in this project, an overview has been made of the services in AWS and Azure.

<b>AWS service</b>	<b>Azure service</b>
EC2	VM
VPC	VNet
IAM	AD & RBAC
Network Firewall	Firewall
Firewall Manager	Firewall Manager

Security Hub	Sentinel
Detective	Sentinel
CloudWatch	Monitor
Cost Explorer	Cost Management and Billing

### 3.4 How did that phase go?

This phase isn't complete yet.

### 3.5 Conclusion

The conclusion of this research is that services EC2, VPC, IAM, CloudWatch and Cost Explorer are needed for setting up the network, security, and management of the environment in AWS. The services Network Firewall, Firewall Manager, Security Hub, and Detective are useful for setting up a more advanced firewall solution with additional security and observation in AWS. For Azure this means that the same services should be used, only with a different naming convention, as stated in the table in the report.

## 4. Sub-question 4:

How can we connect on-premises datacenters to a cloud environment?

### 4.1 Introduction

This is the fourth sub-research question, as stated in the Plan of Action (PvA). This question helps us in the process of creating a solution for the clients of the stakeholder who might want to connect their on-site datacenter to our centralized firewall solution.

### 4.2 Why would we want to connect a datacenter to the cloud?

The aim for this project is to create a centralized cloud environment to which other hybrid cloud solutions can connect. In some cases a client hasn't fully migrated to the cloud, and they still keep an on-site datacenter operational. Because the datacenter supports processes and resources which are located in the cloud, it would be beneficial to connect these datacenters to the secure cloud firewall. The internet security for the datacenters would also be more manageable for the company.

### 4.3 Amazon web services

In the first few sprints the focus of our scope lies on Amazon Web services (AWS). This is the service most clients of Sogeti use in their production environment. Amazon web services offers a service application to connect on site environments to their cloud environments. This is called AWS Direct Connect. This service enables clients to connect their datacenters to the cloud with 1 gigabit or 10 gigabit ethernet connections, however there are much variables to reach these connection speeds. Amazon offers a complete guide on how to solve this problem, but it should be noted that Amazon charges some amounts of money for the connections and data transfers made to the cloud. To create a connection one must first contact Amazon to enable this virtual connection, or let an AWS partner configure this. Once this is done, the client must configure their router. The router should support at Border gateway protocol (BGP) with at least a MD5 authentication. Next Amazon offers guidance on how to set up the AWS virtual connection. This solution from amazon is the most premium one. Its also possible to use different options, but those use traditional like VPN solutions.

### 4.4 Microsoft Azure

Microsoft Azures cloud service also offers a solution to connect a data center to the cloud. The main two solutions are Site-to-site VPN, and Azure express route. Express route requires a private connection from a provider and does not involve public internet. Almost like Amazon's Direct connect. In order to establish a connection to Azure with Express route, it is required to purchase ExpressRoute circuits. One circuit supports only 10 connections, so when you are in need of more connections this is a factor to consider. Another benefit is that Azure EXPRESSROUTE works with configuration through PowerShell. This makes it easier to configure large environments and it could be possible to automate this process. Since Windows operating system is also owned by Microsoft, it is possible to connect and integrate on site Active directory services to the cloud.

### 4.5 Conclusion

The conclusion of this research is that both Amazon and Azure cloud providers provide several well documented solutions for connecting on site locations to their cloud platform. The only thing that I'm missing is some security aspects. It's also worth to consider is that that connecting a datacenter to the cloud will probably increase the clients bill, because both services charge extra money in some way. There must be a specific reason why a client wants to keep their datacenter on site because this option is bringing some costs to the company.



## 5. Sub-question 5:

What can we automate with Terraform/cloudformation?

### 5.1 Introduction

This is the fifth sub-research question, as stated in the Plan of Action (PvA). This question helps us in finding out what we can do with Terraform and Cloudformation, and how we can implement it in our project.

### 5.2 What is Terraform, and for what is it used for?

Terraform is a Infrastructure as a code software solutions. Terraform lets sysadmins create, roll out and manage a cloud or on-premise infrastructure or any other platform it supports. The benefit of using terraform is that the configuration files are in readable human language, and this language is a declarative language. This means that you code the result you want, and not the steps it takes to achieve the result. This works very well in devops workplaces where teams need to be agile and changes need to be made frequently.

### 5.3 What is CloudFormation?

CloudFormation is a Infrastructure as a Code (IaC) tool for creating, rolling out and orchestrating AWS Cloud resources. Cloud formation does almost the same as terraform, but is limited to AWS. This is because CloudFormation is a Amazon product. The most noticeable difference is that CloudFormation makes a distinction between code (i.e., templates) and instantiations.

### 5.4 For what is Terraform/CloudFormation used?

Terraform is mainly used for creating, deploying and provisioning infrastructure resources on site, or in the cloud. For example to create the necessary resources for an application, to create a test environment based on a production environment, or to add or change a security permission for a user. It is also possible to make a change, to scale up or scale down. Terraform also work well with Kubernetes. It can create the environment so that its ready to run an application. For OS related orchestration it is better not to use Terraform, but something like Ansible.

### 5.5 How can we apply Terraform or CloudFormation in our project?

The main power of Terraform lies with the deployment of the creation and configuration of infrastructure. Since our project is in a Amazon Web services environment, we can use Terraform to deploy the firewall and its settings into AWS to quickly test or transform the environment to match a different situation. Using Terraform makes it also easier to edit existing configuration files and keep track of version management for the development. In the case that we need to scale up it becomes as simple to just edit the configuration file and push the update. Because of the fact that we use a AWS environment we might need to consider using cloud formation, but cloud formation is not supported on other platforms.

### 5.6 Conclusion

The conclusion is that Terraform can definitely help our project development. We can quickly set up infrastructure to help develop our proof of concept and it will become easier to make changes to the existing infrastructure. With Terraform we can also make changes to the Aws firewall, but the thing with Terraform is that its power lies within infrastructure only, so if we use any software we need another solution for automatization to make changes to the firewall.

## 6. Sub-question 6:

How to assure that the technology follows the design?

### 6.1 Introduction

This is 6th sub-question, which we had to work on to ensure that the technology follows the design with the most suitable project management technique. It was known that we were going to use Agile for this project. This means that research has been done into the most suitable Agile methodologies.

### 6.2 Why is it important that the technology follows the design?

A design sets clear expectation. This allows realistic project results and deadlines to be established. It also puts the customer at ease and reduces the risk of failure.

If the technology follows the design, there are no surprises, ultimately resulting in a successful and profitable product launch.

### 6.3 The benefits of the right Agile methodology

Choosing the right Agile methodology depends on several factors, such as project goals, team types, organization size and organizational culture. Each methodological framework has its own processes, its own tactics, its own prerequisites, implications, and benefits. This means that not every method can be used equally well for a project.

### 6.4 How did that phase go?

The Proof of concept is a sort of small test prior to the project to test the idea. It exists solely to show that a product concept is both functional and can be developed. The Proof of Concept template for IT projects allows us to present a very clear outline of what your IT project is and what it does. Proof of concept helps us present our idea/project in the best possible way to get the needed approval from stakeholders.

### 6.5 Conclusion

The conclusion of this research is that the Kanban and Scrum method is most suitable as a project management technique. Both methods are suitable for smaller teams to work together effectively. Since in scrum there is no visualization of the workflow and Kanban does not use timeframes. The methods will be combined to have the best of both worlds.

## 7. Sub-question 7:

How can we create a PoC based on the technical design?

### 7.1 Introduction

The Proof of concept is a sort of small test prior to the project to test the idea. It exists solely to show that a product concept is both functional and can be developed. The Proof of Concept template for IT projects allows us to present a very clear outline of what your IT project is and what it does. Proof of concept helps us present our idea/project in the best possible way so as to get the needed approval from stakeholders.

### 7.2 When do you use a proof of concept?

A proof of concept is most valuable when you're developing a new product and method. We will need a proof of concept when:

- Creating a new project idea: If we are developing a product or method that no one has done before, a proof of concept will serve as our pilot project. When you do not have previous use cases for comparison, you'll need to test your idea and ensure it makes sense in real life
- Adding a new feature to a project: When we add a new feature to a project, the existing project becomes something new. If we invented the new feature we are adding to the project, use a proof of concept to ensure the feature will not negatively affect the project's functionality

We may not need a proof of concept to demonstrate feasibility if we are not adding a new idea or feature to our project. We really need to determine whether there's precedence for the project we are working on, and if there is, use data in place of a POC

### 7.3 Benefits of Using Proof of Concept & steps to write a PoC

A POC verifies that concepts and theories applied to a project will result in a successful final product. POCs do not produce deliverables, as the core issue being considered is the feasibility of the project.

Here are some of the main benefits of using PoC during the early development stages of our project:

- ✓ PoC gives the stakeholder valuable data to determine if the project or product idea will be feasible and how valuable it will be for the target audience.
- ✓ PoC gives us important user feedback, and information about market demand, target audience and critical pain points.
- ✓ Proof of concept is an important step of the product development process because it's the base for our product prototype and minimum viable product.

A proof of concept is a project, and like any project, it must be clearly defined. Even though there is no standardized method to write a proof of concept we will need to break down the POC process into actionable steps.

- Define Your Success Criteria - To have accurate proof of concept feasibility measurements, we must have a set of metrics or success criteria. To define the success criteria for our POC process, we start by interviewing the client, as it is their satisfaction that will determine if the proof of concept is a success.
- Estimate Duration & Effort - When you're working on a PoC it's a project, but it's not the final project. You'll need to estimate the duration and effort that will be put into the proof of concept pilot project.
- Define the Scope of our PoC - The scope of our PoC determines what will be done and measured in the PoC pilot project. Defining the scope for our proof of concept is key to getting accurate results. Even if the POC is proved viable, that proof of concept is worthless if the scope is not correct, because we failed to evaluate the project thoroughly.
- Pick our Resources - Who we choose to execute the proof of concept pilot project is as important as the planning process. We want to make sure they have the right skills to do a thorough job.

## 7.4 Comparison between PoC & Prototype & MVP

Proof of Concept	Prototype	Minimum viable Product
Demonstrates feasibility	Demonstrates how to build the product	Uses feedback to make changes to product
Takes days or weeks to create	Takes weeks to create	Takes months to create
Created for developers or researchers	Create for developers, stakeholders, or limited end users	Created for end users
Used to test an idea never created before	Used to get funding by showing how the product will look	Used to look for market fit by assessing user feedback

The proof of concept tests our product or idea at an elementary level, while the prototype brings our idea to life so we can share it with others. The MVP is our prototype in its final stage before we deliver our product or idea to the masses.

## 7.5 How did that process go?

## 7.6 Conclusion

## 8. Sub-question 8:

How can Sogeti efficiently update their firewall rules?

### 8.1 Introduction

One of our main tasks during this project is to find the best possible way to update their firewall in an automated way. This means we need to use Terraform with CI/CD, which will execute the changes to the rules in the most simple way possible. The reason for this is that it will save a lot of time and be extremely efficient.

### 8.2 CI/CD

CI/CD stands for the combination of Continuous Integration (CI) and Continuous Delivery (CD). It is a method that allows developers to automate the software and application delivery process. When developers are constantly changing code, CI/CD allows them to frequently make incremental changes, take part in multi-department collaboration, avoid errors in deployment, increase deployment speed, and automate deployments.

### 8.3 Terraform Cloud

Terraform Cloud is HashiCorp's managed service offering. It eliminates the need for unnecessary tooling and documentation for practitioners, teams, and organizations to use Terraform in production. Provision infrastructure in a remote environment that is optimized for the Terraform workflow

### 8.4 Conclusion

## 9. Sub-question 9:

What are the challenges in designing this solution?

### 9.1 Preface

To complete this project, we need to fulfill a certain amount of requirements. These requirements are based on the wishes and demands of the stakeholder and should result in a technical solution with a proof of concept. However due to some complexity of some topics, some of the requirements will be harder to complete than others. The goal of this research is to identify some challenges ahead we will face during the project.

### 9.2 Requirements

The stakeholder wishes to connect his customers to the remote firewall environment. The firewall should manage the different clouds. The control of the AWS environment should be automated through a commit from GitHub. When some code is committed, the environment should automatically create the appropriate resources and configure it accordingly in a secure manner. The Terraform should manage this process of creating and managing the infrastructure. On top of this we should be able to connect multiple cloud platforms to this environment. With all these technical requirements it can become complex at some point. And at some point we are at risk to get lost in the technical stuff. This is because there could be multiple solutions implemented and there is so much information available.

The main things we need to work on are the pipelines, the security/access control and the automation part of the project. Terraform will take care of the automation, however we are using the Hashicorp version. This version is much simpler, but has less advanced features. The stakeholder prefers the CDK version of terraform. Another point on automation is that It's a requirement from the stakeholder to use separate repositories on the github. This would be 2 repositories total. We have a basic understanding on how to set up the pipelines and how to use them. However to automate from two separate repositories could become a challenge to set up and configure because of some possible limitations.

Another challenge could be to make the process of automation more functional. At this point to create new resources we need to edit a central file in the Github. It would be preferable if we can commit a new file for each new resource, and then the resource gets created. If the file is deleted, the resource would be deleted as well. Constantly editing a single file could become annoying for the administrator and could increase the margin of error.

### 9.3 Conclusion

The conclusion of this report is that most of the technical aspects seem doable by our team. However putting the final touches on the project seem the most challenging aspects at the moment. to tackle this issue we could conduct more research, but we have to keep watch of the time because the stakeholder expects that we deliver a minimal viable product in a short time. To assist us during the resource we could ask for guidance form the tutors and expert who are familiar with these technologies. Also in case that the solution inst all that clear and user friendly, we should not forget to write a good manual for the future users and administrators for our proof of concept to help them tackle these challenges that we faced during the development of the project.