Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

Sigurd Ose Dybdahl
Jonas Dale Fredriksen
Wiktor Miklaszewicz

Kubernetes Platform for Secure Container Migration

Bachelor's thesis in Digital Infrastructure and Cyber Security
Supervisor: Erik Hjelmås
May 2023

NTNU
Norwegian University of
Science and Technology

Sigurd Ose Dybdahl
Jonas Dale Fredriksen
Wiktor Miklaszewicz

Kubernetes Platform for Secure Container Migration

Bachelor's thesis in Digital Infrastructure and Cyber Security
Supervisor: Erik Hjelmås
May 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Kubernetes Platform for Secure Container Migration

Sigurd Ose Dybdahl Jonas Dale Fredriksen
Wiktor Miklaszewicz

May 20, 2023

Abstract

Norsk helsenett, a Norwegian IT company delivering digital infrastructure and services for the Norwegian health sector, sought a solution to establish a seamless Kubernetes platform between their private cloud and a public cloud, Microsoft Azure. The objective was to design a Kubernetes-based cloud solution enabling the movement of containers between private and public cloud environments. Since Norsk helsenett operates and manages sensitive data, we conducted an additional data security risk analysis in order to determine the risk score connected to migrating to a public cloud service primarily based outside of the EU and its strict data privacy laws. The project conducts a comprehensive literature review and practical experiments, focusing on Azure Kubernetes Service, the Schrems II case, and the European Data Protection Board's recommendations for migration to countries outside the EU. The end result is a hypothetical Kubernetes-based migration model, using a blue-green replacement pattern, ArgoCD, Helm charts, and MongoDB. This model, along with our findings in the security part, suggests that a hybrid cloud solution is essential to ensure data confidentiality. The result of the risk analysis suggests that sensitive data should be stored on-premises, while less sensitive data could be suitable for the public cloud with adequate additional security measures.

Sammendrag

Norsk helsenett, et norsk IT selskap som leverer digital infrastruktur og tjenester til den norske helsesektoren, ser etter en sømløs Kubernetes løsning mellom deres private sky og en offentlig sky, Microsoft Azure. Målet var å designe en Kubernetes-basert skyløsning som muligjør flytting av containere mellom private og offentlige skyløsninger. Med tanke på at Norsk helsenett administrerer sensitiv data, har vi også gjort en risiko analyse rundt datasikkerheten for å se på risikoen rundt det å migrere data til en offentlig skyløsning basert utenfor den europeiske union, med sine strenge personvernsregler. Prosjektet er basert på literaturstudier og praktiske forsøk, med fokus på Azure Kubernetes Service, Schrems II-dommen og Det europeiske Personvernrådet anbefalinger for migrering av data til land utenfor EU. Sluttresultatet er en hypotetisk Kubernetes-basert migrasjonsmodell, som bruker ett blue-green replacement pattern, ArgoCD, Helm charts og MongoDB. Denne modellen, i sammarbeid med våre resultater i sikkerhetsdelen antyder at en hybrid skyløsning er nødvendig for å garantere datakonfidensialitet. Risikoanalysen antyder at sensitiv data bør lagres on-prem, altså lokalt, mens mindre sensitiv data kan være egnet for lagring i en offentlig sky med tilstrekkelige sikkerhetstiltak.

Contents

Figures

Tables

Code Listings

Acronyms

AKS Azure Kubernetes Service. ii, iii, 4, 26, 27, 30, 32, 58

AWS Amazon Web Services. 4, 58

CLI command line interface. 17

CNI Container Networking Interface. 30

DGA Data Governance Act. 41, 49

EDPB European Data Protection Board. ii, 1, 6, 22, 38, 41, 49, 58

EU European Union. 7, 21, 38–42, 46

FISA Foreign Intelligence Surveillance Act. 38–41, 49

GDPR General Data Protection Regulation. 22, 38, 39, 41, 42, 58

k8s Kubernetes. ii, iii, 1–6, 9, 12–18, 21, 23–27, 29, 30, 32, 33, 47, 49, 50, 53, 57, 58

NHN Norsk helsenett. ii, iii, 1, 3–6, 9, 11, 21–23, 28, 29, 36, 38–43, 47, 49, 50, 52, 57–59

NTNU Norwegian University of Science and Technology. 2, 3, 5, 7, 11, 26

PV PersistentVolume. 17, 32

PVC PersistentVolumeClaim. 17, 32, 35

SCCs Standard Contractual Clauses. 39

TADPF Trans-Atlantic Data Privacy Framework. 41, 42, 49

Glossary

API An application programming interface. This lets you interact with specific parts of a program from another program. 37

Azure Microsoft's public cloud platform. 4, 5, 13, 25, 26

CI/CD pipeline A combination of continous intergration and continous delivery. 17

continous delivery A development practice where code changes are automatically prepared for delivery. 17, 29

continous intergration Frequent merging of several small changes into a main branch. 17

loose coupling Hardware and software components that interact when necessary. In a loosely-coupled multiprocessing environment, where several computers share the workload, a machine can be added and replaced without shutting down the entire system.. 16

NoSQL A database that is not based on SQL, meaning that the data is not relational. 18

OpenStack Open source cloud computing infrastructure software. 26

replica set MongoDB: A collection of MongoDB instances maintaining the same data set.. 28

replicaset Kubernetes: Maintain a consistent and stable group of active pods continuously.. 15

SkyHiGh NTNU's OpenStack Platform. 26, 27

stateful application An application that saves data to a persistent storage. 16

Chapter 1

Introduction

1.1 Project Background

This bachelor thesis was provided by Norsk helsenett (NHN). NHN is a Norwegian IT company that delivers digital infrastructure and services for the Norwegian health sector. They are responsible for developing, managing, and operating various digital infrastructure and national e-health services. These services include helsenorge.no and the core health journal, all with the aim of improving the overall efficiency of the Norwegian healthcare system.

Norsk helsenett's private cloud is a platform that is built upon software-defined infrastructure. It makes use of VMware Cloud foundation (VCF) and vSphere with Tanzu, so it is able to deliver both traditional and modern applications. The NHN private cloud also deliver a Kubernetes (k8s)-platform where container services can run and internal customers/users can manage their code and distribute their own services. As of today the Kubernetes-platform only runs on NHN's private cloud. Therefore NHN asked us to look at the possibility of running containers from the platform in a public cloud.

1.1.1 Problem Area

The problem that we are trying to solve is to establish a Kubernetes platform on a public cloud that is able to work seamlessly with the private cloud infrastructure of Norsk helsenett, allowing for the migration of container workloads between the two. In addition to this, we will evaluate the security aspect of the data stored in a public cloud based on the security recommendations from the European Data Protection Board (EDPB) and relevant upcoming frameworks.

It is important to mention that the Problem description has changed since the preliminary assumptions. This is based on the challenges we met along the way while trying to create a proof-of-concept. A more in depth explanation can be found in Chapter 1.3.1 Limitations.

1.1.2 Thesis Rationale

The decision to choose this thesis was motivated by several factors. The topic presented a new challenge that none of the group members had worked with before. Two out of three members has basic knowledge of cloud computing, and one of the members has a basic understanding of infrastructure as code. Kubernetes is a relatively new technology that requires extensive research and learning, even for those with some background in related fields [1].

There is a big demand for Kubernetes, which is an exciting technology that is gaining popularity rapidly. Digitalization is happening at an unprecedented pace, and hybrid cloud migration is a relevant topic that needs to be addressed urgently. One of the most critical factors to consider during the migration process is the security of data, given the multiple locations where data can be stored. As of February 2023, after the Schrems II case and invalidation of EU-US Privacy Shield [2], few frameworks secure the confidentiality of data sent outside of the EU.

This thesis aims to contribute to the existing body of knowledge by providing insight into the challenges of hybrid cloud migration and the need for secure data migration practises. We will also present upcoming relevant data privacy frameworks that are currently in the works.

1.1.3 Academic Background

Our bachelor group consists of three students, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Miklaszewicz. We are all in the Bachelors program "Digital Infrastructure and Cybersecurity" at Norwegian University of Science and Technology (NTNU), campus Gjøvik. The courses we took and consider to be relevant for the thesis are:

DCSG1001 - Infrastructure: Basic skills:
Low level Linux commands, basic understanding of hardware, operating systems, virtualization, cloud solutions, computer networks, and the historical development of IT.

IDATG2204 - Data modeling and database systems:
Overall knowledge about databases.

DCSG2005 - Risk Management:
Thorough knowledge on information security management, risk management and governance.

DCSG2003 - Robust and scalable services:
Thorough knowledge of service architectures, their performance, scalability, availability, and security.

INFT2504 - Cloud services as a business:
Thorough knowledge on cloud productivity, collaboration, industry best prac-

tices, and secure implementation.

IIKG3005 - Infrastructure as Code:
Knowledge on IaC principles, professional system administration workflow, and cross-platform administration.

Not all group members have taken the last two courses, therefore an additional skill development phase was included at the beginning, focusing mainly on cloud computing and infrastructure as code§. This allowed all members to have a solid foundation and understanding of the necessary concepts before beginning the main project.

1.1.4 Support Actors

Our contact point from the employer Norsk helsenett (NHN) is Håvard Elnan, senior system engineer. He has a lot of experience in Kubernetes field and was able to provide technical aid when needed.

The bachelor thesis was supervised by Erik Hjelmås, Associate Professor at the Norwegian University of Science and Technology. Erik was able to provide administrative guidance for our overall process.

1.2 Requirements

1.2.1 Time Requirements

This project was estimated to go over an 18 week period. The minimum time required for each group member was estimated to about 30 hours each week, which should result with a total of about 1600 work hours. In case there is an extraordinary demand for extra work hours, the time frame will be adjusted accordingly. [1]

1.2.2 Language Requirements

The thesis is exclusively using English as the writing language. The prevalence of English-language resources, particularly in the fields of Information Technology and Kubernetes, makes it more convenient when it comes to understanding and explaining technical phrases, terms, articles and documentation.

As Kubernetes is an open-source platform, its user community is widely dispersed across the globe and uses English as the primary language of communication. Thus, writing the thesis in English will help to ensure that the research is accessible and understandable to a wider audience users and enthusiasts.

[1] For the exact amount of hours used, see Appendix C

Figure 1.1: NHN's toolkit

1.2.3 Software Requirements

We have a relatively high degree of freedom when choosing the necessary software for our thesis. The main stipulation from the client is that our containers should be managed by Kubernetes and that Azure should serve as our primary cloud storage platform. This, in turn, necessitated our use of Azure Kubernetes Service (AKS).

In addition to Azure Kubernetes Service we were free to use Kubernetes tools from Norsk helsenett's existing toolkit, see Figure 1.1.

1.3 Project Scope

1.3.1 Limitations

In the first meeting with Norsk helsenett, the group was told that the project would be based on both Azure and Amazon Web Services (AWS). Later it was determined that expending time and resources to look at them both, considering they work on similar basis. Thus, it was chosen that the group would focus on just Azure.

It was also decided that taking a look at the pricing model for Azure will not be a part of the bachelor thesis since it is not really relevant for the technical and security aspects of data migration. Thus, pricing will not be looked at unless the group has any extra time at the end of the process.

Version	Release Date	End of Life date
1.21	8 April 2021	28 June 2022
1.22	4 August 2021	28 October 2022
1.23	7 December 2021	28 February 2023
1.24	3 May 2022	28 July 2023
1.25	23 August 2022	27 October 2023
1.26	**9 December 2022**	**24 February 2024**
1.27	11 April 20203	Unknown

Table 1.1: Kubernetes release timeline as of March 2023 [4]

	Old version, end of life
	Old version, still maintained
	Latest stable version
	Future release

Table 1.2: Legend [4]

Norsk helsenett has a toolkit they use for their Kubernetes enviroment called nhn-tooling. This consists of many programs shown in Figure 1.1, including Prometheus, Grafana, Kyverno, Falco and others. This toolkit is dependent on a lot of the infrastructure currently in production. The group does not have access to this infrastructure, and will therefore not be able to use nhn-tooling while working on the project. This will result in our Kubernetes infrastructure having different dependencies and requirements than the one NHN uses. Our solution aims to be a proof-of-concept for NHN, instead of a guide that can be followed step for step.

We have also discovered that the private cloud platforms operated by NHN and NTNU are currently utilizing an outdated version of Kubernetes as of February 2023, version 1.21, that is no longer supported by Azure as of August 2022 [3], see Table 1.1 and 1.2 for Kubernetes version history. As a result, any attempts to transfer data between these private clouds and Azure will be hindered.

While NHN has plans to update to a newer, supported version later this year, the timing of the update unfortunately does not align with our current project timeline. To mitigate this challenge, we will utilize two separate Azure clusters for the proof-of-concept model, with the assumption that NHN will eventually update their cloud and establish the necessary network connections between their private cloud and Azure. For further reasoning see Chapter 4.3.

1.3.2 Target Audience

The main target audience for this thesis is individuals who possess some level of IT experience, such as cloud computing and containerization, and are familiar with Kubernetes but have not yet used it in practice. The knowledge requirement allows the report to avoid explaining the foundational concepts and instead focus on providing valuable insights into the migration process.

In addition to the previously mentioned main target audience, it is worth noting that Kubernetes is an open-source platform that is used widely across different industries. This means that the target audience for this thesis also includes anyone who is interested in utilizing Kubernetes to migrate containers.

1.4 Project Goals

1.4.1 Result Goals

1. Designing a Kubernetes system that enables the movement of containers between private and public cloud environments.
2. Assessing to what degree a public cloud service provider meets the security standards set by the European Data Protection Board.
3. Providing practical recommendations that will help organizations make informed and secure decisions when migrating to the public cloud.

1.4.2 Effect Goals

1. Enhancing the versatility of container hosting options and the ability to choose the location where they are hosted.
2. Improve the overall performance, scalability and cost-efficiency of the NHN infrastructure while meeting the given security requirements.
3. Achieve an acceptable risk level linked to data migration.

1.4.3 Learning Goals

1. Gain proficiency with Kubernetes and related software tools like: ArgoCD, GitOps and Helm.
2. Acquire experience in working collaboratively through being part of a team for an extended period of time.
3. Obtain practical experience through working with an actual client.
4. Acquire a comprehensive understanding of the security risks and threats associated with data migration outside of the European Union.

1.5 Thesis Structure

In order to present our research findings in a clear and organized manner, we have chosen to adopt the IMRaD (Introduction, Methodology, Results, and Discussion) structure for our bachelor's thesis [5]. This structure is well-suited to our approach, since our project is mostly based on finds and not an actual product-development.

We made the decision to include Chapter 3: Background Theory directly after Chapter: 2 Method to ensure that our readers are on a similar theoretical footing as we were after several weeks of research and skill development. By providing this theoretical foundation, we aim to create a strong basis for the rest of the thesis and ensure that the reader is better equipped to comprehend and evaluate our research methods and findings.

The following structure was based on Norwegian University of Science and Technology (NTNU) recommendations for IMRaD structures [6]:

Introduction: Overview of the research problem and objectives.

Method: Methods use to conduct the thesis.

Background Theory: Necessary theoretical foundation.

Practical Implementation: Practical research conducted.

Data Security: Security aspect behind migration of data outside of the European Union (EU).

Discussion: Presentation of results, goals and self-criticism.

Conclusion: Evaluation of the conclusion and further work.

1.6 Risk Analysis on Project Level

Based on the risk analysis conducted in the preliminary project section in Appendix A, the following three risks have been identified as having the highest risk scores, making them the primary concerns for our case.

4. Client not able to give the needed assets and guidance, or there are communication problems:
The group will have to contact the supervisor and come up with an optimal solution. This will most likely result in continuing the project with a smaller scope or redoing the entire assignment.
Mitigation: Clear communication with the client and supervisor will be important to make sure the group has everything they need.
Consequence: 4
Probability: 2
Risk Score: 8

5. Data loss:
Try to restore from a backup. If this is not possible, the data that was lost will have to be redone.
Mitigation: Everyone will be required to take regular backups. There will also be a git repository that overleaf will automatically push to. The group members will pull from this project every couple days to make sure they have a recent backup on their own computer.
Consequence: 4
Probability: 2
Risk Score: 8

9. Scope mismanagement:
If the scope gets too big or too small, the supervisor will be contacted so the project gets back on the right track.
Mitigation: Good communication with the supervisor and the client to make sure the group is on the right track.
Consequence: 4
Probability: 3
Risk Score: 12

Remaining project risks and the final risk matrix can be found in the preliminary project in Appendix A.

1.7 Tools Used for the Project

Toggl Track:

Toggl Track has been used to track the time spent on the different parts of the project.

Trello:

Trello has been used for a Kanban board. For more information about how Kanban was used in this project, see Chapter 2.2. Trello was also used as a way to create the structure for this report by putting the different sections in order to see what would work well after each other. See Figure 1.2 for an example of this.

TikZ and pgfgantt:

The TikZ package is used for making graphics in LaTeX [7]. The pgfgantt package provides the ganttchart environment, which draws a Gantt-chart within a TikZ picture [8]. This has been used to create the different Gantt-charts in this document.

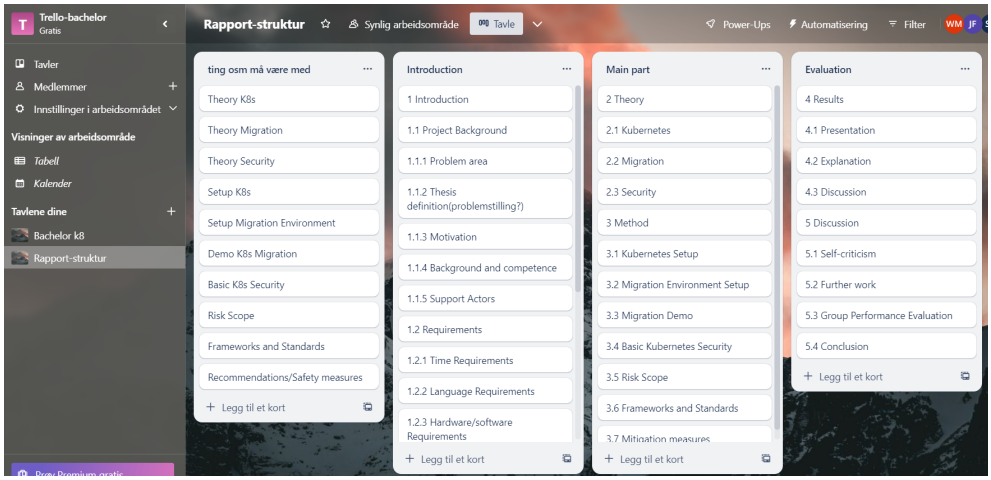


Figure 1.2: Initial structure plan from February 2023

Overleaf:

This thesis has been written in Overleaf, a collaborative LaTeX editor.

Github:

Github has been used as a source code repository.

Microsoft Visio:

Microsoft Visio is a program for making vector graphics. All graphics in this thesis are made in Visio.

ChatGPT 3.0:

ChatGPT 3.0 has been used in this project for grammar correction and rephraseing of sentences that sound unnatural. It has also been used for fixing LaTeX syntax that we did not manage to fix ourself.

Microsoft Teams:

In the start of the project when the group was learning Kubernetes the wiki feature on Teams was used for sharing information inside the group.

Microsoft Teams has also been used for meetings with Norsk helsenett, as well as communication between meetings.

1.8 Summary of Project Phases

Our work process is reflected by the following phases, which correspond to different chapters in the report. The reason for this decision is to present the thesis in a chronological and organized manner, with each chapter building on the knowledge gained from the previous one.

Theoretical skill development and research:

In this phase, we included Chapter 3: Background Theory, which summarizes the theoretical knowledge and skills gained in the first five weeks of the project. At the beginning of this phase, our group had little to no knowledge of the technical and practical aspects of this chapter.

Hand-on skill development and research:

In this phase, we included Chapter 4: Practical Implementation and Chapter 5: Data Security. These two chapters build upon the theoretical background gained in the previous chapter plus the eight next weeks and provide practical knowledge related to the implementation of the migration project and the security of the data involved.

Theoretical summary of combined results:

In this phase, we included Chapter 6: Discussion and Chapter 7: Conlusion. Chapter 6: Discussion serves as a presentation and discussion of the results obtained from the previous phases, and provide a comprehensive summary of our goals. The final stage of our thesis concludes with Chapter 7, which summarizes the entire thesis and provides a conclusive statement regarding our research.

Chapter 2

Method

2.1 Weekly Structure

This section provides insights into how we structured our workdays and coordinated with each other, our supervisor and point of contact from Norsk helsenett. It also outlines the different types of meetings we held, their frequency, and their purpose in ensuring that we remained on track and met project milestones.

Normal workday:

A normal workday consists of working with planned tasks. Most days, we will sit together in meeting rooms at campus. We start each day by booking a room two weeks in advance, which is as long into the future as the NTNU room booking system allows us to. We start the day off at 9:00 in the morning every day and work until anywhere between 14:30 and 16:00, depending on what needs to be done that day. Collaborating on a daily basis ensures that everyone is on the same page, and makes helping each other simpler.

Meeting with supervisor:

Initially, we had a meeting with our supervisor, Erik Hjelmås, every Wednesday at 9:00-9:30. Later, this got changed to 9:30-10:00. This meeting consists of weekly guidance and updates on the group's progress. The majority of feedback on the report is given here.

Meeting with client:

Every Thursday from 13:00 to 14:00, we have meetings with our point of contact from Norsk helsenett, Håvard Elnan. This meeting consists of the same as the meeting with our supervisor, but from a more technical standpoint. In late February, we started having daily standups at 14:00-14:15

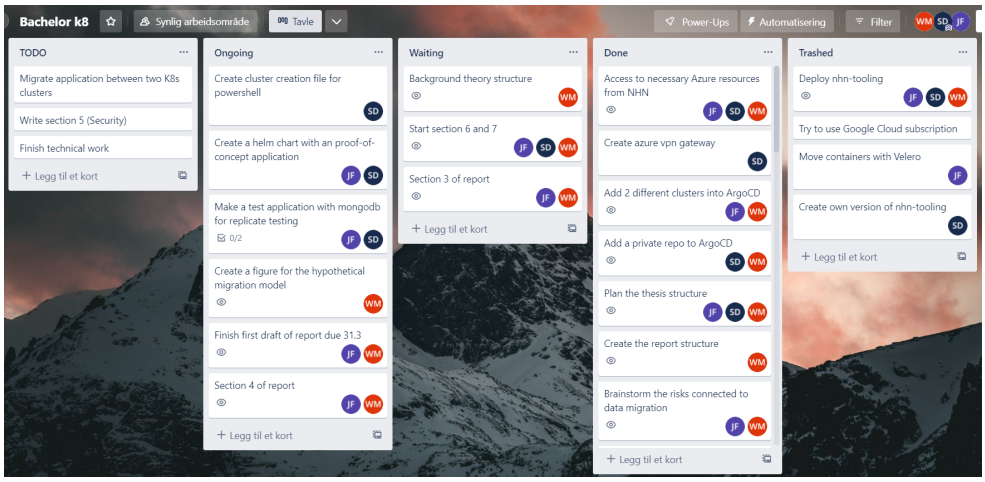


Figure 2.1: Example of our Kanban board

with Håvard in addition to the weekly meeting. We started having these daily meetings after we were stuck on a pretty simple problem in Kubernetes for almost a week, where Håvard gave us a fix in 5 minutes.

Weekly summary meeting:

We had a weekly meeting in the group at 14:00-14:30 every Friday. This meeting consists of summing up the week's work and setting new goals for the next week. This is also where we update our Kanban board.

2.2 Kanban

The group chose Kanban as the method for visualizing and tracking the progress of the project, see Appendix A for in depth reasoning behind the choice to use Kanban. It allows us to see which tasks are pending, which are being worked on, which are waiting for dependencies, and which have been completed. Our Kanban board was initially divided into four categories, but there was a need for a fifth one serving as a thrash bin. The **TODO** category contains tasks that need to be done, while the **Ongoing** category contains tasks that are being actively worked on. The **Waiting** category contains tasks that are waiting on dependencies or for other tasks to be completed before they can be started. The **Trashed** category contains tasks that are no longer relevant or are deemed unnecessary. Finally, the **Done** category contains tasks that have been completed. See Figure 2.1 for an example of our board.

The use of the Kanban method will help the group stay organized and focused on the tasks that need to be prioritized. It also provides a clear overview of the project, which will be helpful in identifying areas that need more attention.

However, because of a lack of constant maintenance we did not use it to its full potential, and the board faded away from the work process. The Kanban board was a valuable tool in the beginning of the project, but the board started to have less value as time went by.

2.3 Literature Review

A literature review involves an in-depth analysis of various sources of information and literature pertaining to a specific research topic [9]. In our case, we conducted a literature study on the topic of Kubernetes and data transfer. To achieve this, we will engagage in multiple Kubernetes tutorials, such as the one provided on the Kubernetes homepage [10]. We will explore various websites and sources for documentation on Kubernetes, and watch a range of videos on YouTube that provide insight on the subject matter.

2.4 Practical Experiments

The practical experiments in this project aims to gain hands-on experience and validate the concepts explored in the literature review. These practical experiments will start off with the Kubernetes tutorials already mentioned. After following these tutorials, the group will create basic services using the different technologies that will be used in the migration theory setup, notably Azure, ArgoCD and Kubernetes, based on the knowledge gained so far. Following this, the main technical work will start, where progress towards a solution for the project will be made. This solution will iterate and evolve based on feedback from the client.

Chapter 3

Background Theory

This chapter provides a summary of the first phase of our project work, and serves as an introduction to the background theory needed in order to comprehend the subsequent sections of the thesis. The goal of this section is to provide the same theoretical foundation that we acquired during the first month of the project.

3.1 Kubernetes

Kubernetes, or k8s for short, is an open source platform for managing containerized workloads and services developed by Google. Kubernetes has been adopted by a large, active community and receives continuous development, with new features and regular improvements. The shorthand k8s results from the eight letters between the "K" and the "s" [1]. Kubernetes is a large and complicated technology, and therefore there is a lot of components and terminology needed to understand. The following sections will present the most important parts of Kubernetes, divided into three sections, Compute management, Container management and Volume management. See Figure 3.1 for a visual representation.

3.1.1 Compute Management

To effectively use Kubernetes, it is essential to have a basic understanding of its components and terminology.

Cluster:

> Software running on Kubernetes, it is executed within a cluster comprised of worker nodes, which are in turn overseen by a master node. The master node is responsible for managing and monitoring the entire cluster [11].

Node:

A physical or virtual machine that is part of a Kubernetes cluster. Each node is responsible for running containerized applications and providing computing resources for those applications.

Control plane:

The set of components that make up the "brain" of a Kubernetes cluster. It includes several components that work together to manage the cluster, including:

API (kubectl):
The Kubernetes API is the interface used to interact with a Kubernetes cluster. It provides a programmatic way to manage the cluster, allowing you to create, update, and delete Kubernetes resources like pods, services, and deployments. The kubectl command-line tool is a way to interact with the Kubernetes API.

Scheduler:
The component that is responsible for scheduling pods (groups of one or more containers) onto nodes in the cluster.

Controller manager:
The component that is responsible for monitoring the state of the cluster and making changes to ensure that the desired state is maintained.

3.1.2 Container Management

The next area of focus encompasses the crucial process of deploying and scaling containers effectively.

Pod:

A pod is the smallest unit of deployment in Kubernetes and contains one or more containers, which share the same network and storage resources, and are scheduled onto the same node.

Deployment:

A Kubernetes deployment is a declarative configuration that defines a desired state for how a pod or replicaset should be deployed and managed within the Kubernetes cluster.

Service:

A service provides a way to expose a set of pods running the same application as a network service, which can be accessed by other parts of the cluster

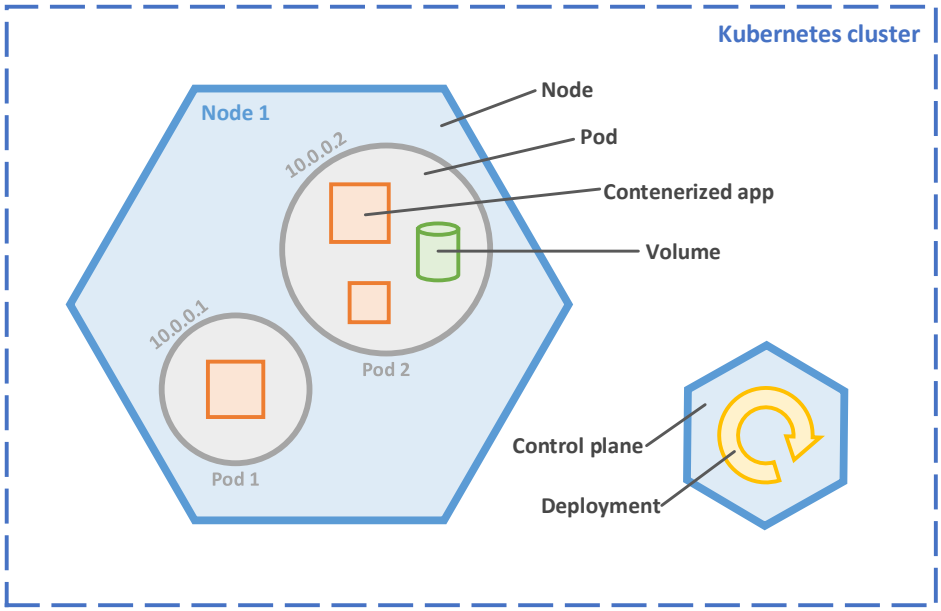


Figure 3.1: Kubernetes environment

or the external network. Services enable loose coupling between dependent parts of an application and can provide load balancing and service discovery.

Namespace:

A namespace provides a way to divide cluster resources between multiple users or teams. It can be used to create logical boundaries between different parts of a application or to allocate resources to different projects or environments.

StatefulSet:

A StatefulSet manages the deployment and scaling of a set of pods, each with a unique identity, that are deployed in a specific order and require persistent storage. StatefulSets can be used to deploy stateful application such as databases or key-value stores, and provide guarantees around ordering and uniqueness of Pods.

3.1.3 Volume Management

The last important area of Kubernetes is understanding how data is stored in a cluster.

PersistentVolume (PV):

PersistentVolume is a Kubernetes resource that represents a piece of storage in the cluster that has been provisioned by a administrator or dynamically provisioned by a StorageClass. PVs can be used as a way to remove the details of how storage is provisioned and consumed from the pod's configuration. They are useful as they can have their own life cycle independent from the pod life cycle.

PersistentVolumeClaim (PVC):

PersistentVolumeClaim is a Kubernetes resource that is used to request a specific amount from a PV. When the PVC is created, Kubernetes will search for a suitable PV to match PVC requirements such as amount of storage space and access mode, then binds them together.

StorageClass:

StorageClass is a Kubernetes object that defines different storage options available in the cluster. When you create a PVC without specifying a specific PV, Kubernetes uses the StorageClass to dynamically provision a PV that matches the PVC's requirements.

3.2 Additional Tools

In addition to Kubernetes, several other tools were utilized in this project, such as ArgoCD, Helm, Bitnami, and MongoDB.

GitOps:

Although GitOps is a method for working, not a tool, it is important to comprehend GitOps before learning about ArgoCD. With GitOps approach the desired state of infrastructure and applications is defined and managed as code stored in a version control system like Git. Changes to the infrastructure are made through pull requests and undergo automated testing before being merged and deployed using a CI/CD pipeline [12].

ArgoCD:

ArgoCD is a Kubernetes continous delivery tool that adopts the GitOps approach to streamline application deployments. It tracks Git repositories for any updates and ensures that the Kubernetes cluster can reflect the desired state of the application. ArgoCD has a web interface and a command line interface (CLI), that enables management and deployment of applications. Additionally, ArgoCD simplifies the process of deploying and monitoring applications on multiple clusters [13].

Helm:

Helm is a package manager for Kubernetes that simplifies the development and deployment phase using Helm charts, a reusable template for Kubernetes [14]. A Helm chart is a bundle of preconfigured files designed to deploy an application or a service. This allows deployment of an entire application with a single command. A changed value in the Helm chart will affect every file that utilizes that value, reducing the risk of overlooking a file [15].

Bitnami:

Bitnami is a provider of prepackaged software for different environments, including Helm charts. This comes with pre-implemented configuration making it customizeable, thus simplifying the deployment of a software [16].

MongoDB:

MongoDB is an open-source, document-oriented NoSQL database. It stores data in JSON-like documents called BSON, allowing for dynamic data modeling. MongoDB is designed to handle large amount of data and offers scalability through horizontal scaling across multiple servers or clusters [17].

3.3 Data Migration

Data migration can pose significant challenges, including downtime, data loss, and security risks. The following Figures 3.2, 3.3 and 3.4 are heavily inspired by those found in Kief Morris' book "Infrastructure as code: managing servers in cloud" [18].

3.3.1 Reasoning

One of the key reasons for data migration in a hybrid cloud environment is to enable the migration of workloads between public and private clouds [19]. This provides organizations with the flexibility to choose the cloud infrastructure that best meets their needs for specific workloads. For example, public clouds may be better suited for workloads that require high scalability and reliability, while private clouds may be better suited for workloads that require strict compliance and security controls. Additionally, being able to seamlessly move containers between clusters in the hybrid cloud is important for ensuring high availability and disaster recovery, as well as optimizing resource utilization and reducing costs [20].

3.3.2 Migration Challenges

One of the main challenges in data migration is ensuring zero-downtime changes during the process [21]. This involves deploying new versions of the application

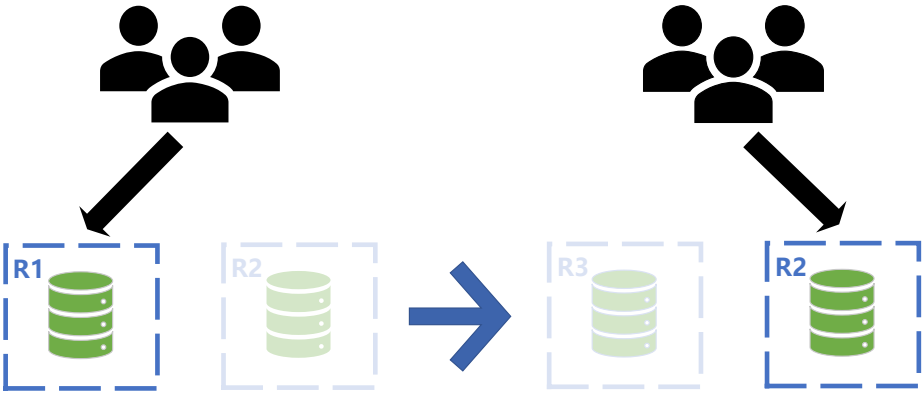


Figure 3.2: Pattern: Blue-green replacement

while minimizing or eliminating interruptions to the end-user experience. One approach to achieving zero-downtime changes is defined in "Infrastructure as code: managing servers in cloud" as using specific migration patterns such as blue-green replacement, phoenix replacement, or canary replacement [18]:

Blue-Green Replacement:

> Figure 3.2 shows a blue-green pattern. It involves deploying a new version of the application to a new environment (green), while keeping the previous version (blue) in the existing environment. Once the new version is verified to be working correctly, traffic can be switched from the old environment to the new one, providing zero-downtime changes. One of the main advantages of this pattern is that it allows for the quick and safe deployment of new versions, without any downtime or impact on users. It also provides a rollback option in case of issues with the new version, as traffic can be easily redirected back to the old environment. However, the downside of this pattern is that it requires duplication of infrastructure, which can lead to higher costs and complexity [18].

Phoenix Replacement:

> Figure 3.3 shows a phoenix replacement pattern. It is a deployment strategy that involves replacing the entire infrastructure with a new version. Unlike blue-green deployment, which switches between two identical environments, the phoenix pattern completely replaces the existing environment. This can be useful when significant changes to the infrastructure are required. One key difference is that there is no option to roll back to the previous environment. The phoenix pattern requires more downtime but can be more efficient in terms of resource utilization [18].

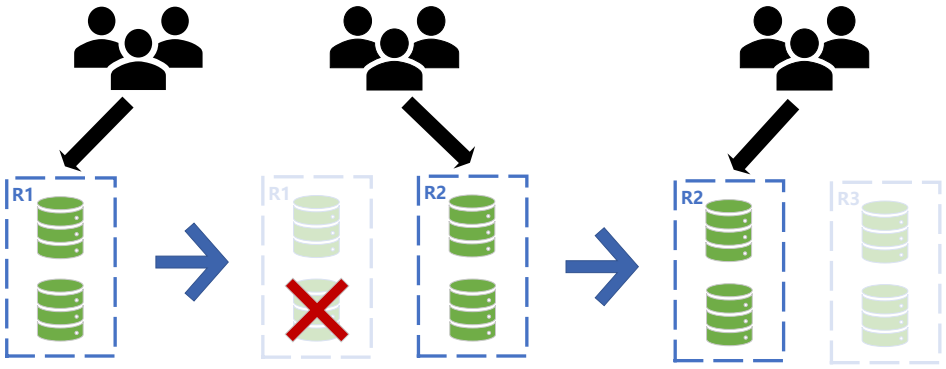


Figure 3.3: Pattern: Phoenix replacement

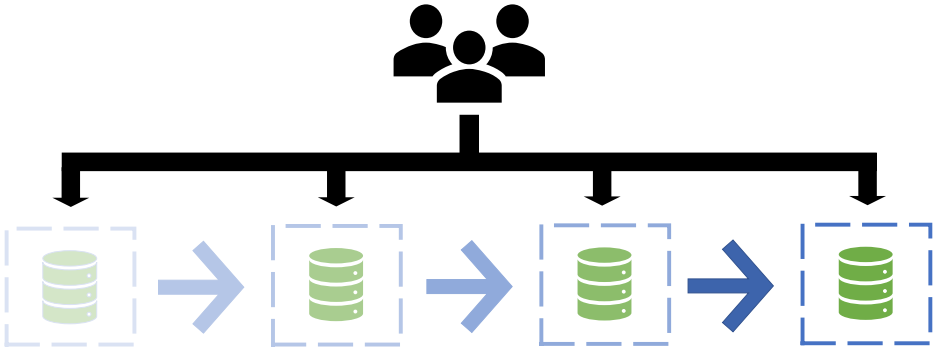


Figure 3.4: Pattern: Canary replacement

Canary Replacement:

> Figure 3.4 shows a canary replacement pattern. This pattern involves deploying a new version of the application to a small subset of users (canaries), while keeping the previous version for the rest of the users. This allows for verifying that the new version is working correctly before rolling it out to all users. Unlike the previous patterns, the old version is not immediately decommissioned, but it is phased out gradually [18].

Other challenges of data migration include ensuring data consistency and integrity during the migration process, minimizing or eliminating data loss, and maintaining application performance and functionality. These challenges can be addressed through careful planning, testing, and monitoring of the migration process.

3.4 Security Background

In the cloud, data is stored off-premises, which creates a challenge for businesses to maintain control over their data, ensure its privacy and security, and comply with regulations.

While software and hardware vulnerabilities are the most common risks in traditional IT infrastructures [22], data privacy is the primary concern in cloud computing [23]. Companies face the task of ensuring that their sensitive data is secure, confidential, and protected from unauthorized access or breaches.

We will examine the various security risks and challenges involved in migrating data to the public cloud and the significance of data privacy in this process. We reached an agreement with the client to disregard Kubernetes' built-in security mechanisms, such as role-based access control, network policies, secret management, or image security. This decision was made because it does not matter how secure the underlying Kubernetes security is if the transfer of data is prohibited due to privacy regulations in the first place.

3.4.1 Reasoning

The focus of our concern over data privacy in a cloud environment stems from a policy change that effected our client, Norsk helsenett (NHN). They originally worked on-prem and wanted to establish their platform on Microsoft Azure. At the time, the client believed that their data was protected by the US's Privacy Shield framework, which came into effect on 12th of July 2016, as a secure bridge for data migration between the European Union and the United States [24]. However, the Schrems II judgement, ruled on the 16th of July 2020 invalidated the Privacy Shield, [2]. The Court of Justice of the European Union in its Case C-311/18 Data Protection Commissioner v Facebook Ireland and Maximillian Schrems (commonly called Schrems II) ruled that:

> *General Data Protection Regulation (GDPR) applies to the transfer of personal data for commercial purposes from a company in the EU to a company in a third country, regardless of whether that data may be accessed by the authorities of that third country for national security purposes. The court also stated that the use of standard contractual clauses for transferring data to a third country must ensure a level of protection essentially equivalent to that guaranteed by the GDPR within the EU. If the clauses cannot be complied with in the third country, and the protection of data required by EU law cannot be ensured by other means, then the competent supervisory authority must suspend or prohibit the transfer. Lastly, the court invalidated the EU-US Privacy Shield decision, finding that it did not provide adequate protection for personal data transferred to the US.*

The text above is a summarized and paraphrased version of the actual ruling [2].

This ruling was mainly driven by the US's Foreign Intelligence Surveillance Act (FISA) number 702, which permits the government to conduct targeted surveillance of foreign persons located outside the United States with the assistance of electronic communication service providers to acquire foreign intelligence information [25].

Paragraphs 180 and 181 of the Schrems II case justified that the Privacy Shield could not ensure a satisfactory level of protection for data coming from outside the EU due to the lack of limitations on the power conferred by Section 702 of the FISA, stating the following:

> *It is thus apparent that Section 702 of the FISA does not indicate any limitations on the power it confers to implement surveillance programmes for the purposes of foreign intelligence [...] that article cannot ensure a level of protection essentially equivalent to that guaranteed by the Charter [...]*
>
> *According to the findings in the Privacy Shield Decision, the implementation of the surveillance programmes based on Section 702 of the FISA is, indeed, subject to the requirements of PPD-28. However, although the Commission stated [...] that such requirements are binding on the US intelligence authorities, the US Government has accepted, in reply to a question put by the Court, that PPD-28 does not grant data subjects actionable rights before the courts against the US authorities. Therefore, the Privacy Shield Decision cannot ensure a level of protection essentially equivalent to that arising from the Charter, contrary to the requirement in Article 45(2)(a) of the GDPR that a finding of equivalence depends, inter alia, on whether data subjects whose personal data are being transferred to the third country in question have effective and enforceable rights.*

As a result, Norsk helsenett reverted everything back to on-premise from Azure. We aim to investigate if this still applies in 2023 and if there are any ways to partially migrate data to Azure while ensuring GDPR compliance.

3.4.2 Security Challenges

Since the Schrems II ruling in July 2020 which invalidated the EU-US Privacy Shield, one must look to alternative legal mechanisms for data transfers. The European Data Protection Board (EDPB) has published guidance on assessing the adequacy of third country laws and practices related to surveillance and data access by public authorities [26]. All European companies must ensure that their third-party cloud service providers, such as Microsoft Azure in this case, also comply with the requirements of the General Data Protection Regulation (GDPR) [27].

This can be challenging as a client may not have direct control over the practices of their service providers.

The COVID-19 pandemic has highlighted additional security concerns related to data transfers, as many employees are now working remotely and accessing sensitive data from outside the office. This has increased the risk of data breaches and unauthorized access to data, particularly when data is being transferred across borders. [28]

3.5 First Migration Model

The hypothetical first migration model is based on the theoretical skill development and research phase. This model serves as a starting point for our further analysis and development of an optimal migration strategy. It is essential to note that the model does not comprise any security measures as it served as a reference for our proof-of-concept, only demonstrating a theoretical instance of container migration.

The zero-downtime patterns discussed earlier, blue-green replacement, phoenix replacement, and canary replacement, can all be used alongside Kubernetes and ArgoCD to achieve seamless transitions during data migration. Our evaluation of these patterns led us to select the blue-green replacement pattern due to its considerable advantages compared to others.

The blue-green replacement pattern effectively eliminates downtime, a crucial aspect for applications that require continuous availability. Considering Norsk helsenett manages health care services that needs constant availability, zero downtime is essential. It facilitates scalability by enabling the creation of multiple environment instances, thereby optimizing resource utilization. In the event of eventual deployment issues, the rollback process is straightforward, allowing for easy issue resolution.

While phoenix replacement and canary replacement patterns can be effective in specific scenarios, they do come with certain limitations. Phoenix replacement involves the creation of an entirely new environment for each deployment, which can be time-consuming and resource-intensive. The canary replacement pattern involves gradual traffic routing to the new environment, resulting in a longer transition time that may not be suitable for applications requiring immediate changes.

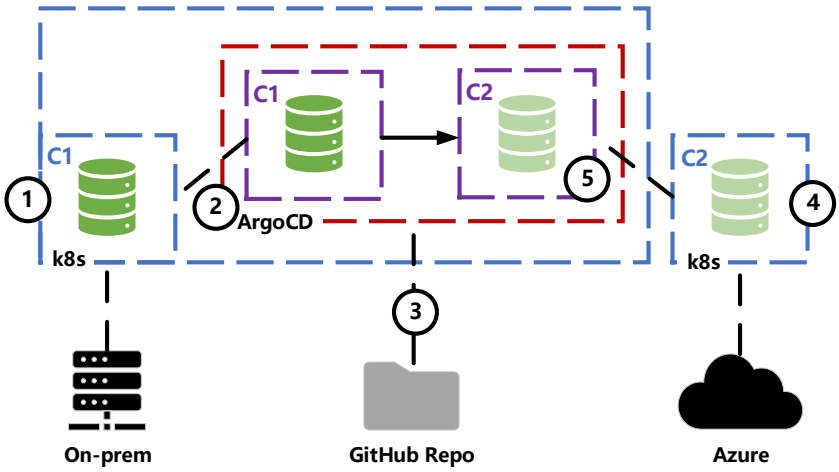


Figure 3.5: First hypothetical migration model

The hypothetical migration model in Figure 3.5 is heavily inspired by Ryan Cook's presentation of GitOps at the LISA19 conference in 2019 [29], further discussed in Chapter 4.2.

For clarity in the thought process, the approach has been divided into six distinct parts, enumerated as follows. This is a theoretical speculation based on one month's worth of research.

1. Creation and deployment of Cluster 1 with Application 1 on-premises.
2. Deployment of ArgoCD on Cluster 1's namespace, followed by the addition of Application 1.
3. Connection of the GitHub repository containing Application 1 to ArgoCD.
4. Creation and deployment of Cluster 2 with Application 1 on Azure.
5. Addition of Cluster 2 to ArgoCD and synchronization of the applications to ensure consistency across both clusters.
6. Turn off Cluster 1 ensuring seamless migration.

Our goal is to successfully migrate the application between two clusters using the previously mentioned concepts and technologies. By achievieng this goal, we aim to demonstrate the feasibility of our theoretical approach and gain insights into the potential for scalability and consistency for data migration using Kubernetes and ArgoCD. The model will serve as both a hypothesis and a baseline for further research and testing.

Chapter 4

Practical implementation

The practical phase of our project builds upon the research and learning phases. This chapter aims to provide a transparent account of the practical work, detailing the extent to which the initial assumptions and hypothesis evolves as we progress towards a functional Kubernetes (k8s) platform for container migration in a hybrid cloud environment.

4.1 Our Introduction to Kubernetes

When the project started, the plan was to spend two weeks learning the theoretical background necessary to complete the project, with the practical work beginning in the third week. It quickly became apparent that this was a underestimation. None of the team members had any prior experience with Kubernetes and the basic theoretical work required to understand what Kubernetes is and how to use it took three weeks.

The initial three weeks of theory was not enough to begin the work on the project. To address this, an additional week of basic Kubernetes tutorials and exercises on the Kubernetes website was added [10].

4.2 USENIX and ArgoCD

USENIX is an organization that supports advanced computing systems communities and promotes innovative research through conferences and publications [30]. At the LISA19 conference, a presentation was held that demonstrated how an application could be moved between three different Azure clusters using ArgoCD [29]. This is similar to what the team is planning to do on a larger and more complex scale.

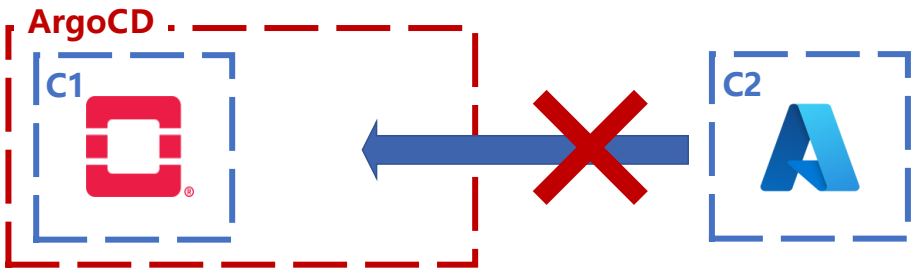


Figure 4.1: Attempt to reach Openstack from AKS

For testing ArgoCD manifest files, a simple application is created and uploaded to a Git repository. The application is retrieved from the repository by ArgoCD, which automatically updates it on the SkyHiGh cloud platform when changes are made to the code and a new version is uploaded to the repository. For example, ArgoCD creates an additional replica automatically when the replica count is changed from two to three. Despite some challenges with understanding how to use ArgoCD and Kubernetes, the test project is successful on SkyHiGh, and later on Azure as well.

4.3 Azure and SkyHiGh

In the previous Chapters 3.1 and 3.2, the practical experiments with Kubernetes were initially conducted on Norwegian University of Science and Technology (NTNU)'s own cloud computing platform, SkyHiGh, which is built on OpenStack [31]. However, several challenges were encountered while working with SkyHiGh, which led to an exclusive switch to Azure for the project.

The original hypothesis was to use SkyHiGh as a simulation of an on-prem cloud solution and then integrate it with Azure and its Azure Kubernetes Service (AKS) as the public cloud solution. However, it was soon realized that OpenStack was not designed primarily for Kubernetes resulting in several problems throughout the beginning phases of the project.

SkyHiGh worked well for creating local Kubernetes platforms for relatively small tests, however issues arose when attempting to connect the SkyHiGh OpenStack cluster with the AKS cluster in ArgoCD as illustrated in Figure 4.1.

Due to SkyHiGh's access restrictions, it is not possible to establish a direct connection with an AKS cluster without a VPN. SkyHiGh is isolated from the external network by default, and difficulties were faced when attempting to connect it with the AKS cluster in ArgoCD. After some discussion the group concluded that it might be possible to reach Azure from SkyHiGh, but as illustrated in Figure 4.2 it did not work

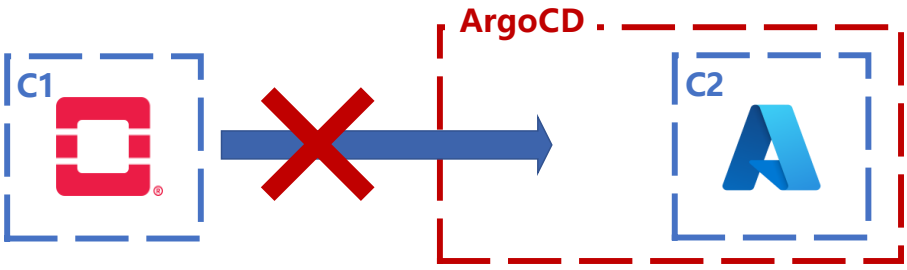


Figure 4.2: Attempt to reach AKS from Openstack

The Kubernetes versions have to be the same in order for the two clusters to work together in ArgoCD [32]. SkyHiGh's Kubernetes was set to version 1.21 at this time, which according to official Kubernetes site is outdated and no longer supported by ArgoCD or AKS, see Table 1.1 [4]. An attempt was made to downgrade the AKS cluster to version 1.21, but Azure will not allow the use of versions older than 1.24 and automatically updates legacy versions to up-to-date ones [3].

In the short message exchange with SkyHiGh's administrator Lars Erik Pedersen, it was found out that the OpenStack Magnum release NTNU uses, which is 13.0.0 (Xena) has an outdated Kubernetes version and can only support up to Kubernetes version 1.21 which is not maintained as of 28th of June 2022. There are plans of updating to Release 14.0.0 (Yoga) in the near future, which supports Kubernetes up to version 1.23 [33]. However, version 1.23 has an end of life date of 28th of February 2023, which is only a week after access could be gained to Openstack Magnum version 14.0.0. As a last resort, Lars Erik tried to set up Kubernetes version 1.24 in a test platform, but with no success:

> *Not sure if this will work unfortunately. I tested myself both in SkyHiGh, and in the test platform (which now supports v1.23.x), with v1.24.10, but it doesn't seem to work very well. It seems as you have to set something up on your own without using the Openstack integrated solution.*

The decision was made to switch exclusively to Azure as the main cloud platform for the proof-of-concept. This resolves many of the issues faced, particularly with regards to network connectivity between the two clusters. Focusing exclusively on Azure allows attention to be directed towards creating a seamless migration environment without being encumbered by the technical challenges of network connection between multiple cloud platforms.

However, the team faced a challenge as the Azure licenses had not been received from the client yet. To proceed, a free trial account was created, offering a limited credit of 200$ and 30 days of usage.

4.4 Our Migration Theories

The migration theory is focused on containerized database migration. Database migration is an important and complex task that can pose significant challenges for software applications [21]. The migration process requires careful planning and execution to ensure that the new environment can function as a complete replacement for the old one, or co-exist with the old environment to increase scalability and redundancy. To achieve this goal, a systematic approach is necessary to ensure a smooth and seamless migration process while minimizing potential disruptions and downtime.

The theory focuses on deploying a new containerized database on a different cloud platform and making it a part of the replic set with the old database. ArgoCD deploys the database and provides continuous changes to the new environment. This approach ensures that the new environment remains up-to-date with the latest configurations changes while minimizing the risk of human error.

During the replication process both the old and the new database can be utilized to ensure data consistency and avoid disruptions. This approach allows for a seamless transition to the new environment by using a blue-green switch to direct network traffic from the old database to the new one. In the event that something goes wrong with the switch to the new database it is always possible to switch the directing traffic back to the old database, thus minimizing the risk of data loss or service disruption.

4.5 Database and Deployment Tool Setup

MongoDB:

In addition to Norsk helsenett already using MongoDB, it also fulfills the requirement of a database being able to be deployed as a replica set, which means that the data can be replicated on multiple processes that maintain the same dataset [34]. We are using the free version, which has limitations that we will later realize are important. [1]

Helm:

When deciding on a Helm chart to use for our MongoDB setup, there are specific requirements that need to be met. The most important requirement is the ability to be configured as a replica set. Another requirement is that it needs to be possible to customize and change the default configuration values to suit the specific needs of our environment. The Bitnami/MongoDB

[1] When we neared the end of our work with MongoDB we were not able to finish it. We suspect this is because of the limitations of the free version or limitations to the Bitnami/MongoDB chart, but we are not sure. See Chapter 4.8 for more info.

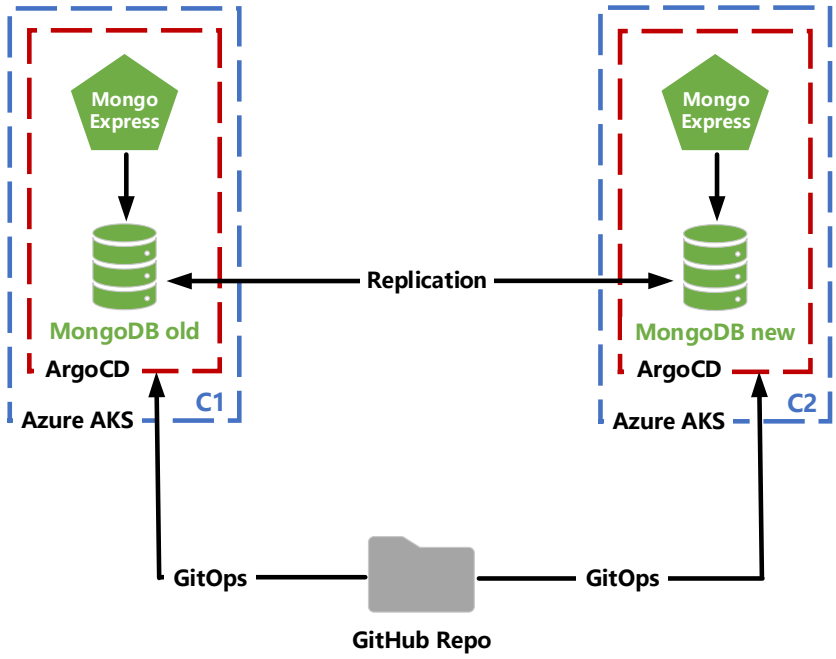


Figure 4.3: Theory with chosen database

chart used meets these requirements and provides a customizable way to deploy MongoDB in a Kubernetes environment.

MongoExpress:

To facilitate testing and demonstrate an application running on top of the database MongoExpress is used. MongoExpress is a web-based administrative interface for MongoDB. MongoExpress can be used both as an application in front of the database and as a tool for administering MongoDB, making it easier to implement test data and gain insight in the database. See Figure 4.3 for a visual representation of how MongoExpress is applied in our improved hypothetical model that includes MongoDB and MangoExpress.

ArgoCD:

NHN already utilizes ArgoCD in their own systems as a continous delivery tool. The decision to use ArgoCD was based on the opportunity to provide practical learning of GitOps and continous delivery. In addition, implementing ArgoCD will speed up the testing process as it makes sure the applications are always up to date with the Git repository.

4.6 Kubernetes Setup

The setup is a standard Azure Kubernetes Service setup. The reason for going for a standard setup is to not include any technical overhead that might intervene with trying to recreate this setup. For the cluster preset configuration we went for the **`Cost-optimized`** preset with a **`B2s`** virtual machine size. The B-series virtual machines are meant for workloads that do not need the full performance of the CPU continuously. It normally runs on 40% CPU performance, but can for periods of time accelerate up 200%. This makes it cheap for the migration setup testing as it does not require a high performance over a longer period of time. The pricing for CPU usage is measured by [35]:

$$\frac{\text{Base CPU performance of vm} - \text{CPU Usage}}{100} * 60 \text{ minutes}$$

It also sets the maximum node count to one and turns off auto scaling to further limit the cost of the Kubernetes clusters.

When deploying an AKS cluster the network configuration can be chosen as either Kubenet or Azure Container Networking Interface (CNI). The main difference between Kubenet and Azure CNI is that Kubenets gives the pods and the clusters logically different address spaces, while Azure CNI assigns pods and clusters IP-addresses within the same virtual network. This means that Kubenet also applies a NAT between the different networks unlike Azure CNI. Azure CNI also offers greater configuration options [36]. For the proof of concept the Azure CNI configuration will be set as it has the option to add both clusters to the same virtual network, as this will simplify the network configuration.

To deploy ArgoCD on a Kubernetes cluster, it is necessary to make a namespace called **`argocd`**. To deploy it on the newly made namespace, the preconfigured file **`install.yaml`** created by ArgoCD is used.[2] In order to deploy the repository for the migration environment setup, it is important that the context is set to the namespace **`argocd`**. Otherwise, Kubernetes will not find the ArgoCD configuration map. The Code listing 4.1 shows the deployment of ArgoCD.

```
kubectl create namespace argocd
kubectl apply -n argocd -f install.yaml
kubectl config set-context --current --namespace=argocd
argocd repo add git@github.com:dybsi/Bachelorrepo.git --ssh-private-key-path bagitkey
```

Code listing 4.1: ArgoCD deployment commands

[2] https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

4.7 Migration Environment Setup

The Bitnami/MongoDB chart comes with preimplemented configuration options that can be utilised through a value file. The **`mongodb-primary.yaml`** value file shown in Code listing 4.2, shows the configuration of the primary(old) database when deploying with the Bitnami/MongoDB chart. It specifies that the architecture should be a replica set, which is essential for the setup. Authentication is enabled, and the replica set key is the same as the secondary database since both will be part of the same replica set. External access is disabled because a different load balancer service will be used instead of the one provided by the Bitnami/MongoDB chart.

```
architecture: replicaset
replicaCount: 1
auth:
  enabled: true
  rootPassword: secret-root-pwd
  replicaSetKey: replicakey
persistence:
    storageClass: "standard"
fullnameOverride: "mongodb-pri"
externalAccess:
  enabled: false
  service:
    type: LoadBalancer
    autoDiscovery:
      enabled: true
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal: "true"
rbac:
    create: true
arbiter:
  enabled: false
primarydb: true
```

Code listing 4.2: mongodb-primary.yaml

The value file **`mongodb-secondary.yaml`** shown in Code listing 4.3 is similar to **`mongodb-primary.yaml`**, but instead uses the options **`primarydb`** and **`primarydbhost`** to set the primary database shown in Code listing 4.2 as its primary database.

```
architecture: replicaset
replicaCount: 1
auth:
  enabled: true
  rootPassword: secret-root-pwd
  replicaSetKey: replicakey
persistence:
    storageClass: "standard"
fullnameOverride: "mongodb-sec"
externalAccess:
  enabled: false
  service:
    type: LoadBalancer
    autoDiscovery:
      enabled: true
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal: "true"
```

```
rbac:
    create: true
arbiter:
  enabled: false
primarydb: false
primarydbhost: "primary-mongodb:27017"
```

Code listing 4.3: mongodb-secondary.yaml

The Bitnami/MongoDB chart need to create a StorageClass and a PersistentVolumeClaim (PVC) to provide persistent storage and is created by the **`storageclass.yaml`** file, which can be seen in Code listing 4.4. This makes it so the MongoDB data is stored on the persistent storage rather than the pods itself. Because pods are ephemeral, meaning they get easily destroyed and recreated, storing data on the pods is not a good idea. Storing the data on a persistent storage will ensure that the data persists even if the pod is terminated or recreated. The storage class determines the type of storage that will be used for creating persistent volumes dynamically. Specifically, the volume will be created using **`disk.csi.azure.com`**, which is a driver for provisioning persistent storage on AKS using the Azure Container Storage Interface (CSI). However, the CSI driver requires additional parameters to be set. One such parameter is the **`SkuName`**, which is a unique identifier for different pricing tiers of Azure services. In this case, the **`SkuName standard_LR`** indicates the use of standard locally redundant storage in Azure. Another parameter, **`storageAccountType`**, specifies the type of storage account to be used. In this case, the chosen type is **`Standard_LRS`**, which provides locally reduntant storage. To ensure that the data is not lost when the PVC's are deleted, the **`ReclaimPolicy`** is set to **`retain`**, which allows the persistent volume to persist and be reclaimed when needed.

The PersistentVolumeClaim requests 2 Gigabytes of storage using the **`standard`** storage class. The access mode is set to **`readWriteOnce`**, which means that the volume can be mounted for reading and writing by a single node at a time.

Once the PVC is created, Kubernetes will dynamically create a PersistentVolume (PV) that satisfies the PVC's storage requirements and binds it to the PVC. The application can then use the PVC to access the persistent storage for storing data.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: standard
provisioner: disk.csi.azure.com
parameters:
  skuName: Standard_LR
  storageAccountType: Standard_LRS
reclaimPolicy: Retain
allowVolumeExpansion: true
mountOptions:
  - debug
volumeBindingMode: Immediate
```

```
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-mongodb
spec:
  storageClassName: standard
  accessModes:
   - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
```

Code listing 4.4: storageclass.yaml

Code listing 4.5 shows an added part of the premade file **`statefulset.yaml`** from the Bitnami/MongoDB chart. It determines whether the current cluster should act as the primary database or not, based on a boolean value **`primarydb`** from the value files. If it is the primary database, it sets the clusterIP network domain name as the value for the environment variable **`MONGODB_INITIAL_PRIMARY_HOST`**. Otherwise it takes a value from the values file, **`primarydbhost`** which should be the IP or the domain name of the primary database of the primary database cluster.

```
    {{- if .Values.primarydb }}
    - name: MONGODB_INITIAL_PRIMARY_HOST
      value: {{ printf "%s-0.$(K8S_SERVICE_NAME).$(MY_POD_NAMESPACE).svc.%s" (include "
           mongodb.fullname" .) .Values.clusterDomain }}
    {{- else }}
    - name: MONGODB_INITIAL_PRIMARY_HOST
      value: {{ .Values.primarydbhost | quote }}
    {{- end }}
```

Code listing 4.5: statefulset.yaml line 237-243

In order for the database containers to replicate with each other they need to be able to communicate and access each other. However, by default, the databases can only access the internal Kubernetes cluster network, because the Bitnami/-MongoDB chart only creates a service type clusterIP for the pods. The clusterIP exposes the service on the internal Kubernetes cluster network, meaning that they cannot communicate with containers outside of the cluster. To expose the containers to each other, a different Kubernetes service type needs to be attached to the containers. The service object type **`loadbalancer`** is a service object that is used to expose containerized applications to external traffic.

The Bitnami/MongoDB chart have preconfiguration for the creation of a load balancer by default. This came with disadvantages such the need for a list of IP-addresses that can be delegated to services. That seems unnecessary to have in this setup, therefore it was decided to implement a custom file for the creation of external access.

The file **`lbsvc.yaml`** (load balancer service), seen in Code listing 4.6, will create a service type **`loadbalancer`** for each replica of the database, which lets the databases communicate outside of their clusters. The load balancer is applied using a iterative method for Helm charts called **`range`** which works as a foreach loop that counts toward the number **`replicacount`** and deploys a load balancer service on each replica. AKS will by default set the load balancer as a public load balancer and allocate it with a public IP-address. By adding **`azure-load-balancer-internal`** to **`metadata.annotations`**, AKS will instead delegate a IP-address from the virtual network **`metadata.annotations`** is used to attach metadata to objects to provide additional information to other components in the cluster, like the other annotation used to specify a DNS record in the Azure DNS zone.

```
{{- $fullName := include "mongodb.fullname" . }}
{{- $replicaCount := .Values.replicaCount | int }}
{{- $root := . }}

{{- range $i, $e := until $replicaCount }}
{{- $targetPod := printf "%s-%d" (printf "%s" $fullName) $i }}
{{- $_ := set $ "targetPod" $targetPod }}
apiVersion: v1
kind: Service
metadata:
{{- $name := printf "%s-%d-lb" $fullName $i}}
  name: {{ $name | quote}}
  annotations:
     service.beta.kubernetes.io/azure-load-balancer-internal: "true"
     service.beta.kubernetes.io/azure-dns-label-name: {{$name | quote}}
spec:
  type: LoadBalancer
  ports:
     - port: {{ $root.Values.externalAccess.service.ports.mongodb }}
       targetPort: mongodb
  selector: {{- include "common.labels.matchLabels" $ | nindent 4 }}
    app.kubernetes.io/component: mongodb

---
{{- end }}
```

Code listing 4.6: lbsvc.yaml

The migration setup will be deployed using ArgoCD as an application. Each database will have its own application definition file, such as **`app-mongodb-pri.yaml`** shown in Code listing 4.7. In this file, the **spec.source** section specifies the repository and path from where the Helm chart and value file should be fetched. The **`destination server:'https://kubernetes.default.svc'`** indicates that the application will be deployed on the current cluster. Furthermore, the application definition ensures that any changes made in the repository will automatically synchronize and update the deployed resources.

`replicacount`

```
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
```

```
  name: mongo-pri
spec:
  destination:
    name: ''
    namespace: default
    server: 'https://kubernetes.default.svc'
  source:
    path: mongodbbit/mongodb
    repoURL: 'git@github.com:dybsi/Bachelorrepo.git'
    targetRevision: HEAD
    helm:
      valueFiles:
      - mongod-primary.yaml
  sources: []
  project: default
  syncPolicy:
    syncOptions: []
    retry:
      limit: 2
      backoff:
        duration: 5s
        maxDuration: 3m0s
        factor: 2
    automated:
      prune: false
      selfHeal: false
```

Code listing 4.7: app-mongodb-pri.yaml

4.8 Migration Demo

To use the migration environment, first deploy the storageclass and PVC so that the database has a volume to store the data. Then deploy the ArgoCD application definition to the cluster. When this is deployed ArgoCD will fetch the chart from the repository and deploy it to the cluster. This also has to be done on the other cluster as well with **`app-mongodb-sec.yaml`**, see Code snippet 4.8.

```
kubectl apply -f storageclass.yaml
kubectl apply -f app-mongodb-pri.yaml
```

Code listing 4.8: Setup of environment

As of today with the current migration setup, the MongoDB databases cannot be added to each others replica set. The error message received is that the replicasetID is different. In an attempt to troubleshoot this issue, one approach was to reset the database with the incorrect replicasetID. The provided Code listing 4.9 demonstrates an example of trying to reconfigure the replicasetID. The code uses the **`reconfig`** command from MongoDB, to reconfigure the secondary database with the replicasetID of the primary database. However, this approach did not work.

```
rs.reconfig({
   _id: 'rs1',
```

```
   members: [
      {
         _id: 0,
         host: 'mongodb-sec-0.mongodb-sec-headless.default.svc.cluster.local:27017',
         priority: 1,
         votes: 1
      },
      {
         _id: 1,
         host: '10.224.0.223:27017',
         priority: 5,
         votes: 1
      }
   ],
    "settings": {
    "replicaSetId": ObjectId("64426ad0e135c32a2389a600")
}})
```

Code listing 4.9: Reconfig replicaID

We addressed this problem with our point of contact from Norsk helsenett, Håvard Elnan. It was concluded that the issue may be that we are using MongoDB open source community edition instead of MongoDB enterprise. The reason for this suspicion emerges from the fact that as of May 2023, MongoDB enterprise is still in beta for replication between two replicaset with MongoDBmulti [37]. Our other theory is that the solution is not supported by the Bitnami/MongoDB chart or MongoDB image provided by the Bitnami/MongoDB chart. Because of this we do not have any solution on how to finish the practical part of the thesis with the current setup, thus only providing a hypothetical model.

Chapter 5

Data Security

5.1 Security Risks in a Public Cloud Environment

This section will discuss the main security risks associated with storing data in a public cloud environment, with a specific focus on Microsoft Azure as the cloud provider, since that is our client's desired platform. As a US-based organization, Microsoft Azure is subject to US data privacy laws, which are generally perceived to be less stringent than those in the European Union [38]. This highlights the potential impact of weaker data privacy laws on the security of sensitive data. We have prioritized the risks discussed in this section based on their potential high impact on data privacy and security.

1. Data breach:

One of the risks of storing data in a public cloud environment is the potential for data breaches [39]. These can occur due to various reasons, such as vulnerabilities in the cloud infrastructure, insecure APIs, or human error. In Azure, as with any cloud service, clients and users must rely on the cloud provider's security measures, which can lead to a loss of control over their data's security.

2. Lack of control:

As mentioned, cloud service providers are responsible for the security of the underlying infrastructure, which includes the physical data centers, hardware, and network components. However, this also means that clients have less control over their data's security, such as how and where it is stored, and who can access it. When using Azure, users must rely on Microsoft's data centers' security measures and trust that their data is adequately protected by competent personnel.

3. Compliance and legal challenges:

When storing sensitive data in a public cloud environment, there are often compliance and legal requirements that must be met. As mentioned in Chapter 3.4.1, in the United States the Foreign Intelligence Surveillance Act (FISA) allows the government to access data stored by US based companies [25]. This could lead to legal and compliance issues for Norsk helsenett, since they manage sensitive data. Similarly, the EU's General Data Protection Regulation (GDPR) requires that personal data is processed securely and lawfully [40], which can be challenging when using third country public cloud providers.

4. Insider threat

Insider threats refer to the risk of an employee or a trusted third-party accessing or mishandling sensitive data. This risk is not specific to cloud environments but is often exacerbated in public cloud environments due to the shared infrastructure and less control over data. Azure has a wast server infrastructure spanning over all the continents with some of the locations being quite suspicious based of their political situation [41].[1] A rogue employee or a partner with access to Norsk helsenett's data could potentially access or misuse it. Especially in today's remote work-style, escalated by the COVID19 pandemic, which as mentioned in Chapter 3.4.2 had an impact on the overall risk of unauthorized access to sensitive data [28].

5.2 Security Recommendations From EDPB

One of the most critical aspects of ensuring data security is complying with relevant regulations and frameworks. The European Data Protection Board (EDPB) is one such regulatory body that has issued guidelines on data protection for cross-border transfers of personal data outside the European Union (EU). These guidelines are particularly relevant for Norsk helsenett, since they operate in the EU and are looking to utilize public clouds to store or process data.

The European Data Protection Board was created to make sure that organizations in the European Union follow the data protection rules set out in the General Data Protection Regulation when transferring personal data outside the EU. To help organizations comply with these rules, the EDPB issued recommendations, which offer guidance on the additional measures that should be taken to safeguard personal data during such transfers. The Recommendations propose six steps that should be followed when intending to transfer personal data outside the EU [26]:

[1] We are aware that it is possible to choose the location of where the data is stored, but it is still considered as an uncertainty, since the data is still managed by the same corporation.

Step 1: Know the transfer

> *The first step is to identify the personal data being transferred and the reason for the transfer. Organizations need to know the data flows involved and the third countries to which the data is being sent. They also need to consider the type, volume, and frequency of the data transfers, as well as the potential risks to the rights and freedoms of the data subjects [42].*

Norsk helsenett are looking at the possibilities of transferring personal data to Microsoft Azure. The data flow will involve regular and frequent transfers of a significant volume of data to servers located mainly in the United States and Europe.

Step 2: Verify the transfer tool(s)

> *The second step is to verify that the transfer tool being used provides adequate protection for personal data. Organizations should ensure that the tool is valid and up-to-date, and covers all relevant transfers of personal data. They should also consider any additional requirements or obligations under the GDPR that need to be fulfilled.*

Norsk helsenett can use Microsoft's Standard Contractual Clauses (SCCs) as the transfer tool to ensure adequate protection for personal data. The Standard Contractual Clauses have been updated in 2021 and are valid under the GDPR:

> *According to the General Data Protection Regulation (GDPR), contractual clauses ensuring appropriate data protection safeguards can be used as a ground for data transfers from the EU to third countries* [43].

However, Norsk helsenett should ensure that the SCCs cover all relevant transfers of personal data.

Step 3: Assess the law of the third country

> *The third step is to assess the laws and practices of the third country to which the personal data is being transferred. This includes looking at any relevant legal frameworks, such as surveillance laws, and the access that authorities in the third country have to personal data. Organizations should also consider any risks to the rights and freedoms of data subjects, such as the potential for discrimination, harassment, or other forms of abuse.*

The US laws on data protection and privacy are not equivalent to those in the EU, and there are concerns about the government's access to personal data. It is especially visible when the Foreign Intelligence Surveillance Act 702 is taken into consideration, as mentioned in Chapter 3.4.2. Norsk helsenett should consider any risks to the rights and freedoms of their subjects. It is

important to mention that Azure has implemented a number of measures to ensure compliance with US laws such as FISA 702. Such as a Transparency Hub that allows customers to track and monitor government requests for data access and strong encryption measures to protect data at rest and in transit. However, it is still important to note that even with these measures in place, the potential risk of government access to data still exists.

Step 4: Identify and adopt supplementary measures

The fourth step is to identify and adopt additional measures to ensure that personal data is adequately protected during transfers outside the EU. This could include technical or contractual measures, such as encryption or data protection agreements. Organizations should consider the effectiveness of the measures in relation to the specific transfer and the laws and practices of the third country. They should also evaluate whether the measures are feasible and sustainable, and whether they are compatible with the transfer tool(s) being used [42].

Microsoft Azure provides several supplementary measures, including encryption at rest and in transit, Azure Active Directory for authentication and authorization, and access controls to restrict user access to data. However, it is important that Norsk helsenett checks whether these measures are effective in relation to their specific transfer and the laws and practices of the United States.

Step 5: Procedural steps

The fifth step is to implement any procedural steps necessary to ensure that the additional measures are effective. This could include updating data protection policies and procedures, training staff, or conducting audits or risk assessments. Organizations should also consider any contractual or legal obligations that need to be fulfilled, such as informing data subjects about the transfer or obtaining their explicit consent.

Norsk helsenett should inform their clients and customers before proceeding with the transfers.

Step 6: Re-evaluate at appropriate intervals

The final step is to regularly review and update the transfer mechanisms and supplementary measures to ensure that they continue to provide an adequate level of protection for personal data. Organizations should consider any changes to the laws or practices of the third country, or any new risks to the rights and freedoms of data subjects, that may require additional or different supplementary measures.

> Norsk helsenett should actively monitor the laws or practices of the United States. As well as any new risks to the rights of data subjects that may require additional or different supplementary measures. It is especially important that they closely monitor the upcoming data privacy frameworks and laws.

In the case of Norsk helsenett, while Microsoft Azure provides several supplementary measures, NHN needs to carefully evaluate whether these measures are effective in relation to the specific transfer and the laws and practices of the United States, especially FISA 702. As the EDPB Recommendations emphasize, ensuring the consistent application of data protection rules across the EU requires careful consideration of the measures that organizations must take to safeguard personal data when transferring it outside the EU.

However, the evolving nature of cross-border data transfers means that Norsk helsenett must also remain vigilant about new frameworks and regulations that may affect their data protection and security strategies. As such, it is important to stay up to date with changes in data protection laws and to adapt the practices accordingly.

5.3 Upcoming Frameworks

In addition to the European Data Protection Board (EDPB)'s recommendations, there are upcoming frameworks that are expected to have a significant impact on data protection and security in the context of cross-border data transfers. These frameworks include the European Commission's Data Governance Act [44] and the Trans-Atlantic Data Privacy Framework [45].

Data Governance Act

The European Commission's Data Governance Act (DGA) is a proposed regulation that aims to facilitate data sharing across the European Union and increase trust in data intermediaries. The act seeks to create a European single market for data that is open, fair, and secure. It includes provisions for the creation of "data intermediaries" that would facilitate data sharing between companies and organizations, as well as for the establishment of a European Data Innovation Board to promote the use of data for the public good.

One of the key features of the DGA is its focus on ensuring the security and protection of personal data. For example, data intermediaries would be required to comply with the General Data Protection Regulation (GDPR) and take appropriate security measures to protect personal data. The act also includes provisions for ensuring the confidentiality of business data and trade secrets.

The Data Governance Act entered into force on the 23rd of June 2022 and, following a 15-month grace period, will be applicable from September 2023 [46].

Trans-Atlantic Data Privacy Framework

The Trans-Atlantic Data Privacy Framework (TADPF) is another upcoming framework that aims to provide a legal basis for transatlantic data transfers. The framework is being negotiated between the European Commission and the US Department of Commerce, and is intended to replace the Privacy Shield framework, which was invalidated by the European Court of Justice in 2020, as mentioned in Chapter 3.4.2.

The new framework will include provisions for ensuring the protection of personal data during cross-border transfers, as well as mechanisms for resolving disputes and enforcing compliance. It is expected to be based on the EU's General Data Protection Regulation (GDPR) and will require US companies to comply with GDPR principles and provide adequate protection for personal data transferred from the EU to the US.

One of the key goals of the Trans-Atlantic Data Privacy Framework is to ensure that there is a strong legal basis for transatlantic data transfers. The framework is expected to provide greater legal certainty for companies that transfer personal data across the Atlantic, as well as for data subjects whose personal data is transferred.

As of today (25th of April 2023) the TADPF is still in the discussion phase but the European Data Protection Board has approved the draft:

> *Given the concerns expressed and the clarifications required, the EDPB suggests these concerns should be addressed and that the Commission provides the requested clarifications in order to solidify the grounds for the Draft Decision and to ensure a close monitoring of the concrete implementation of this new legal framework, in particular the safeguards it provides, in the future joint reviews [47].*

Compared to the current state of data protection and security, these upcoming frameworks, if implemented, will introduce a more harmonized and consistent approach to data protection and security in the context of cross-border data transfers. They will also provide greater legal certainty for Norsk helsenett and other organizations that aims to transfer personal data across borders, as well as for customers whose personal data will be transferred.

ID	Sensitivity Level	Description
1	Open	Data that is meant to be accessible to the general public and does not contain any sensitive information. It can be freely shared without any concern for privacy or security.
2	Internal	Data that is intended for use by employees within Norsk helsenett. It may contain confidential or proprietary information that should not be disclosed to the public, but it is not considered to be highly sensitive.
3	Protected	Data that is more sensitive than internal data and requires additional security measures to prevent unauthorized access.
4	Heavily Protected	Data that is the most sensitive type of data and requires the highest level of security. Access to heavily protected data is strictly controlled and limited to only those with a need-to-know.

Table 5.1: Data sensitivity description table

5.4 Risk Analysis

A risk analysis is a critical process that aims to identify, evaluate and prioritize potential risks associated with a particular action or decision. In this case, Norsk helsenett (NHN) is considering storing their data in a public cloud, specifically Microsoft Azure. This risk analysis is intended to identify the impact the risks mentioned earlier in Chapter 5.1 has on the different sensitivity levels of data stored by Norsk helsenett. It aims to determine whether storing these different types of data falls under an acceptable or unacceptable risk.

The probability, consequence and risk tables in this section are based of the Norwegian University of Science and Technology's risk assessment, but have been modified to suit our specific needs [48]. The probability intervals for the frequency are determined by the risk's chance of happening in a one year interval.

Data Sensitivity scale

The scale in Table 5.1 is based on Norsk helsenett's own data sensitivity scale obtained by consulting a senior system engineer. It shows the four different levels of data sensitivity and its corresponding ID and description.

ID	Degree of probability	Probability description	Frequency interval(P)
4	**Extremely Likely**	More then fifty percent chance in a year's time	$50\% < P$
3	**Very likely**	Ten to fifty percent chance in a year's time	$10\% \leq P \leq 50\%$
2	**Somewhat likely**	One to ten percent chance in a year's time	$1\% \leq P < 10\%$
1	**Not very likely**	Less then one percent chance in a year's time	$P < 1\%$

Table 5.2: Probability description table

ID	Degree of consequence	Consequence description
4	**Extremely consequence**	The organization will face extremely economic and legal consequences
3	**High consequence**	The organization will face moderate economic and legal consequences
2	**Some consequence**	The organization will face only some economic consequences
1	**No consequence**	The organization will not face any consequences

Table 5.3: Consequence description table

Probability description

Table 5.2 shows the different degrees of probability with corresponding ID, description and frequency interval.

Consequence description

Table 5.3 shows the different degrees of consequence with corresponding ID and description.

Risk table

Table 5.4 shows numbers ranging from one to four with addition of *Criticality - a* and *Risk - b* under *Criticality* and *Risk*, representing the four levels of data sensitivity, and their corresponding criticality and risk.

The formula for calculating the *Risk* in the following tables is:

$$\text{Probability} * \text{Criticality} = \text{Risk}$$

Risk matrix

Table 5.5 shows the risks placed based on their corresponding results from Table 5.4 in a risk matrix using colours shown in Table 5.6 to visualize the outcome.

The acceptable risk is ≤ 3 (green), based on the fact that there is no place for uncertainties when it comes to data privacy. In our opinion there is only acceptable and unacceptable risk, and nothing in between.

ID	Risk Name	Probability	Criticality				Risk			
			1a	2a	3a	4a	1b	2b	3b	4b
1	Data breach	3 [49]	1	2	3	4	3	6	9	12
2	Lack of control	2 [50]	1	1	3	4	2	2	6	8
3	Compliance and legal challenge	4 [25]	1	1	3	4	4	4	12	16
4	Insider threat	1	1	2	3	4	1	2	3	4

Table 5.4: Risk table

Extreme likely	3.1 , 3.2		3.3	3.4
Very Likely	1.1	1.2	1.3	1.4
Somewhat likely	2.1 , 2.2		2.3	2.4
Not very likely	4.1	4.2	4.3	4.4
	No consequence	**Some consequence**	**High consequence**	**Extreme consequence**

Table 5.5: Risk matrix

Colour		Description
Red		Unacceptable risk. Data must not be transferred outside of EU.
Green		Acceptable risk. Data can be transferred outside of EU with adequate additional measures, such as a thorough assessment using EDPB's Recommendations.

Table 5.6: Risk matrix legend

The risk matrix in Table 5.5 shows clearly that data with sensitivity level of three (Protected) and four (Heavily Protected) falls under the unacceptable risk level in each of the four risks. Thus, showing a clear pattern indicating that this particular information should not be stored in a public cloud provider located outside of the European Union such as Azure.

Chapter 6

Discussion

6.1 Result Interpretation

The main problem we aimed to solve in this thesis was to establish a Kubernetes platform on a public cloud that can seamlessly work with the private cloud infrastructure of NHN, enabling the migration of container workloads between the two. However, we encountered several challenges during the proof-of-concept development phase, leading us to modify the original problem description.

Our primary result is a hypothetical migration model shown in Figure 6.1, which is a corrected and expanded version of our original hypothesis shown in Figure 3.5. The final hypothesis is based on our research and practical work, demonstrating the final setup for our proof-of-concept. This final migration model is still just a hypothesis, because we did not manage to complete our technical work, so we can not prove that it will work.

In the practical part of our thesis we created an proof-of-concept migration environment setup. Our solution is a Kubernetes cluster running ArgoCD connected to a repository for seamless GitOps operation. We also modified and extended a pre-configured Bitnami Helm chart to deploy the necessary components for an old and a new MongoDB with configuration to make them work as a primary and secondary database. They are also connected to a persistent storage, as well as MongoExpress on top of the databases. Even though our setup did not work as intended, it is still a fully functional setup that can be used for further work as other project and use cases.

In the data privacy part of the project, our security analysis found that based on NHN's four sensitivity levels of data, most of the data fell under the unacceptable category. *Protected* and *Heavily Protected* data are unsuitable for storage in Azure. *Open* and *Internal* data could be stored in the public cloud, but additional security measures must still be considered.

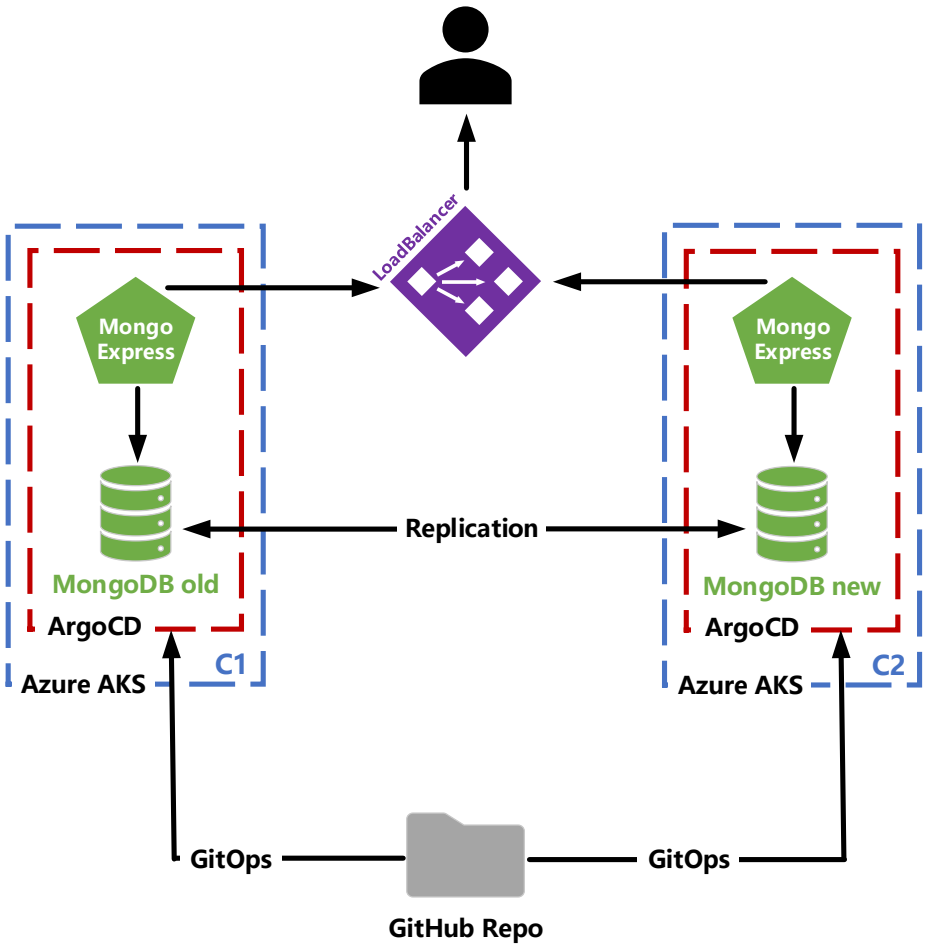


Figure 6.1: Final theoretical hypothesis

6.2 Did We Meet Our Goals?

Result 1: Designing a Kubernetes system that enables the movement of containers between private and public cloud environments.

We have designed a Kubernetes system that enables movement of data between a public and a private cloud environment. However, we do not move entire containers. That was never the main goal from our client, and just a misunderstanding on our part on how Kubernetes worked.

While we have designed this system, we have not tried it in practice, so we do not know for sure that it actually works. This is a subject for further work.

R2: Assessing to what degree a public cloud service provider meets the security standards set by the European Data Protection Board.

We were not able to provide a quantitative assessment of Azure's compliance with European Data Protection Board's security standards, due to the lack of available data and mismanaged scope. However, we were still able to draw conclusions based on our analysis of the recommended six steps. The analysis revealed that Azure does meet many of the security standards set by EDPB. However, we also identified areas where there is a some doubt when it comes to meeting the security recommendations. For instance, we discovered that Azure's primary location, the United States, is subject to laws such as Foreign Intelligence Surveillance Act (FISA) 702. These laws can potentially compromise data confidentiality by allowing the US government to access it. While Azure has implemented measures to try to ensure compliance with similar laws, it is important to be aware of these potential risks when considering the use of Azure or other public cloud service providers.

R3: Providing practical recommendations that will help organizations make informed and secure decisions when migrating to the public cloud.

We examined the current recommendations set by the European Data Protection Board and presented the upcoming frameworks that will have a significant impact on public cloud data transfers outside of the EU. Our analysis showed that the EDPB Recommendations provide valuable guidance on how Norsk helsenett can ensure compliance with data protection laws when migrating to the public cloud. However, we did not create any new recommendations of our own. Instead, we presented the current recommendations from the EDPB, such as the six-step approach to assessing the data protection risks associated with the use of public cloud service providers. Furthermore, we highlighted the upcoming frameworks that will impact public cloud data transfers outside of the EU, such as the Data Governance Act and the Trans-Atlantic Data Privacy Framework, which is an intended replacement for the invalidated EU-US Privacy Shield.

Effect 1: Enhancing the versatility of container hosting options and the ability to choose the location where they are hosted.

We were unable to achieve this goal as we were not able to complete the technical work required. Despite this, we remain confident in the validity of our working theory. With additional effort and expertise, specifically from an individual well-versed in Kubernetes and MongoDB, we are optimistic that this goal can be successfully completed.

E2: Improve the overall performance, scalability and cost-efficiency of the NHN infrastructure while meeting the given security requirements.

This goal was poorly constructed. This project is about moving container data from a private cloud to a public one, as well as the security requirements around this. Theoretically having some containers running on a public cloud can improve performance because you can use better hardware instead of buying it yourself, as well as improving scalability by giving the containers more resources or replicating the containers. As mentioned in Chapter 1.3.1, after discussion with our client in the start of the project we decided that taking a look at the pricing would be outside our original scope, and that we would take a look at it at the end of the project if we had the time, which we did not.

E3: Achieve an acceptable risk level linked to data migration.

This goal was also poorly constructed by us, as we later realized that it was not appropriate to aim for achieving an acceptable risk level linked to data migration. This was due to scope mismanagement and our desire to do more than we realistically were able to. In reality, what we did was to identify the risks connected to the different sensitivity levels of data and determine if they are appropriate to be stored in the public cloud provider Azure. Through our research, we identified the potential risks associated with data migration and evaluated their likelihood and impact on the security of the data based on corresponding sensitivity levels of data being migrated. Although our initial goal of achieving an acceptable risk level linked to data migration was not met, we found out that only four out of 16 risks fell under the category *acceptable risk*, as shown in Table 5.5. Based on these findings, we determined that data from category *Protected* and *Heavily Protected* is not suitable for migration outside of the European Union.

Learning 1: Gain proficiency with Kubernetes and related software tools like: ArgoCD, GitOps and Helm.

At the start of the project we had absolutely no knowledge on the mentioned technologies, or cloud computing in general. While working with Kubernetes for the duration of this project we have learned a lot, but we are by no means experts. As we have mentioned multiple times in this report, Kubernetes is a complicated technology, especially if one have no prior experience with cloud computing. We severely underestimated the complexity of the project, but still learned a lot, so while we are not experts we have gained valuable knowledge.

L2: Acquire experience in working collaboratively through being part of a team for an extended period of time.

Although all group members had prior experience working in groups on various projects, this was the first time we had the opportunity to work together in such a structured manner for an extended period of time. The fixed time schedules helped to cultivate a more work-like mindset, allowing us to operate efficiently and effectively. Despite all group members being enrolled in the same bachelor's program, each of us had different elective subjects and interests within the field of Digital Infrastructure and Cybersecurity. Our shared background provided a solid foundation for collaboration, while our diverse interests and areas of expertise helped us gain valuable experience working collaboratively as a team over an extended period. Our experience working together in a structured manner has allowed us to develop essential skills and expertise in working together on complex projects.

L3: Obtain practical experience through working with an actual client.

Working with an actual client was an invaluable learning experience that provided us with practical skills that cannot be gained through theoretical learning alone. Throughout this project, we had the opportunity to work with a real client and address their needs, requirements, and concerns in a professional and practical manner. One of the significant challenges of working with a client was learning to communicate effectively with them. It was essential to understand their perspective, listen to their needs, and respond appropriately to their concerns. Another key learning experience was project management. Working with a real client required us to manage our time effectively, plan and organize the project, weekly meeting and daily stand-ups. We learned to be flexible and adaptable, able to pivot when requirements or deadlines changed, and to communicate any changes effectively with the client. Working with a client allowed us to apply the theoretical knowledge we had gained in the subjects to a real-world situation. We were able to gain practical experience in problem-solving, communication, and project management.

L4 Acquire a comprehensive understanding of the security risks and threats associated with data migration outside of the European Union.

Acquiring a comprehensive understanding of the security risks and threats associated with data migration outside of the European Union was a crucial learning goal in our bachelor thesis journey. The EU has strict data privacy laws, and as data migration continues to increase, understanding the security risks and threats is essential. One of the significant challenges we faced was understanding the various security risks and threats associated with data migration outside the EU. We had to read and try to understand legal documents, which are often written in a difficult to understand way for someone that has no legal background. We also had to learn about the regulatory frameworks in place in different countries outside the EU, mainly in the United States. Furthermore, we had to consider the legal

implications of data migration outside the EU, such as the General Data Protection Regulation (GDPR) and its requirements for data transfer to third countries. Acquiring a comprehensive understanding of the security risks and threats associated with data migration outside the EU required in-depth research, analysis, and interpretation of various regulatory frameworks and data privacy laws. We gained a wide perspective on today's data privacy landscape and discovered a whole new cybersecurity field for us, which is data migration outside of the European Union and the major security challenges it provides.

6.3 Self-Criticism

Our team encountered some challenges while working on the project. The workload was unevenly distributed, with one team member handling most of the technical work. The reason for this was that while we all started doing different parts of the technical work, the new tasks we gradually needed to do was natural for him to work on because it was closest to what he was already doing. Because of this, it became very hard for the others to catch up, which resulted in one team member taking on most of the technical work. We are unsure if this had any huge impact on our work. While this made the group dependant on one person doing the coding, which could have turned out extremely bad if anything happened, it also let the two other group members get a head start on the main report which was great. Although this was not intentional, and it worked out fine, this should have been prevented by planning better.

However, one person doing the technical work also meant we only needed to have one active Azure subscription, which was a huge bonus. As we have briefly touched on in Chapter 4.3, it took NHN a long time to get us Azure subscriptions. Luckily, Azure has two different free trials, one that gives you 200$ free credits [51], and another student trial that gives you an additional 100$ [52]. We all quickly used up about 50% of the original 200$. This was the point where the technical work started being piled on one person unintentionally. This made his trial run out faster than the rest. However, the other two group members did not need their remaining 100$, so we transferred the work over to another trial and kept working. After all three 200$ trials ran out, we started one student trial, and when that one ran out we started the next one. This let us finish the rest of our work with the three remaining 100$ trials without NHN having to provide us with a subscription. This let us mitigate risk number 4 in our preliminary project plan, "Client not able to give the needed assets and guidance", mentioned in Chapter 1.6.

We are uncertain whether this distribution of tasks was beneficial or detrimental. On the one hand, one person doing most of the technical work made it challenging for others to assist, especially in the theoretical event where he got sick or injured, which luckily did not happen. On the other hand, this approach allowed the rest of the group to begin working on the report earlier and also made managing the free Azure trials much easier.

We started off the report strong, but the intensity of our work gradually decreased as we had to wait for the technical work to be completed. This could have been prevented if the technical work had been more evenly distributed. We faced obstacles in the form of new technologies that we had never used before. These technologies, Kubernetes in particular, were complicated, and we struggled to make progress due to our lack of experience. As a result, we spent a significant amount of time trying to solve problems caused by our unfamiliarity with the technology. The project being too difficult was never considered in our preliminary project plan, which in hindsight it should have been, considering we were working with completely new technology. We did want a new experience, and found Kubernetes interesting, but we might have bit of more than we could chew.

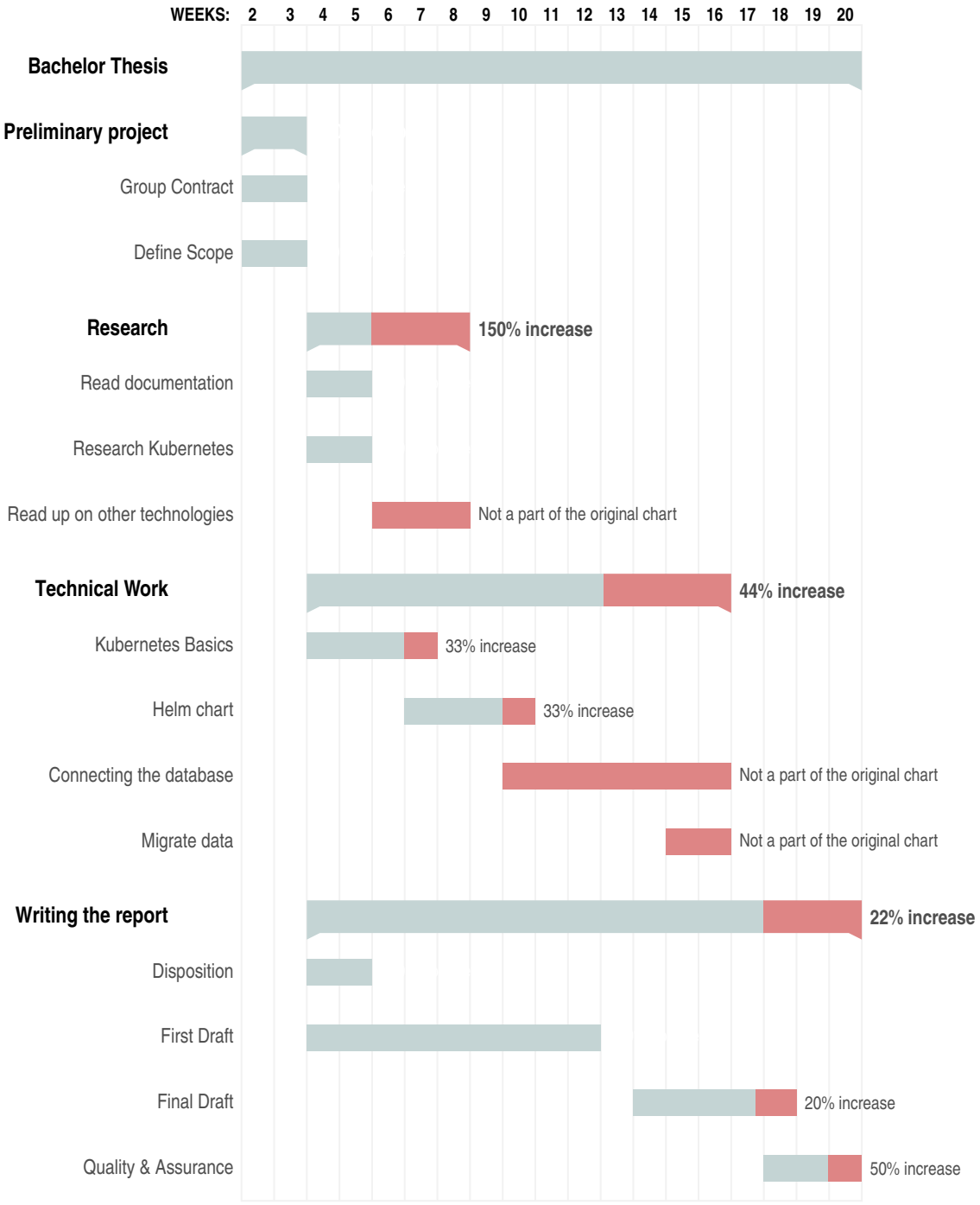
Overall, we learned an important lesson from this experience: it is crucial to distribute the workload evenly and to make sure everyone are caught up on the technical work. As for the problem of the technology being unfamiliar and complicated, we already knew that our project would be about Kubernetes back in November 2022, so we could potentially have started learning Kubernetes back then, before the Bachelor project started. This would have made the start of the project faster, and we could have jumped straight into the more complicated work, instead of spending six to eight weeks learning the basics of Kubernetes.

6.4 Planned vs. Actual Project Progress

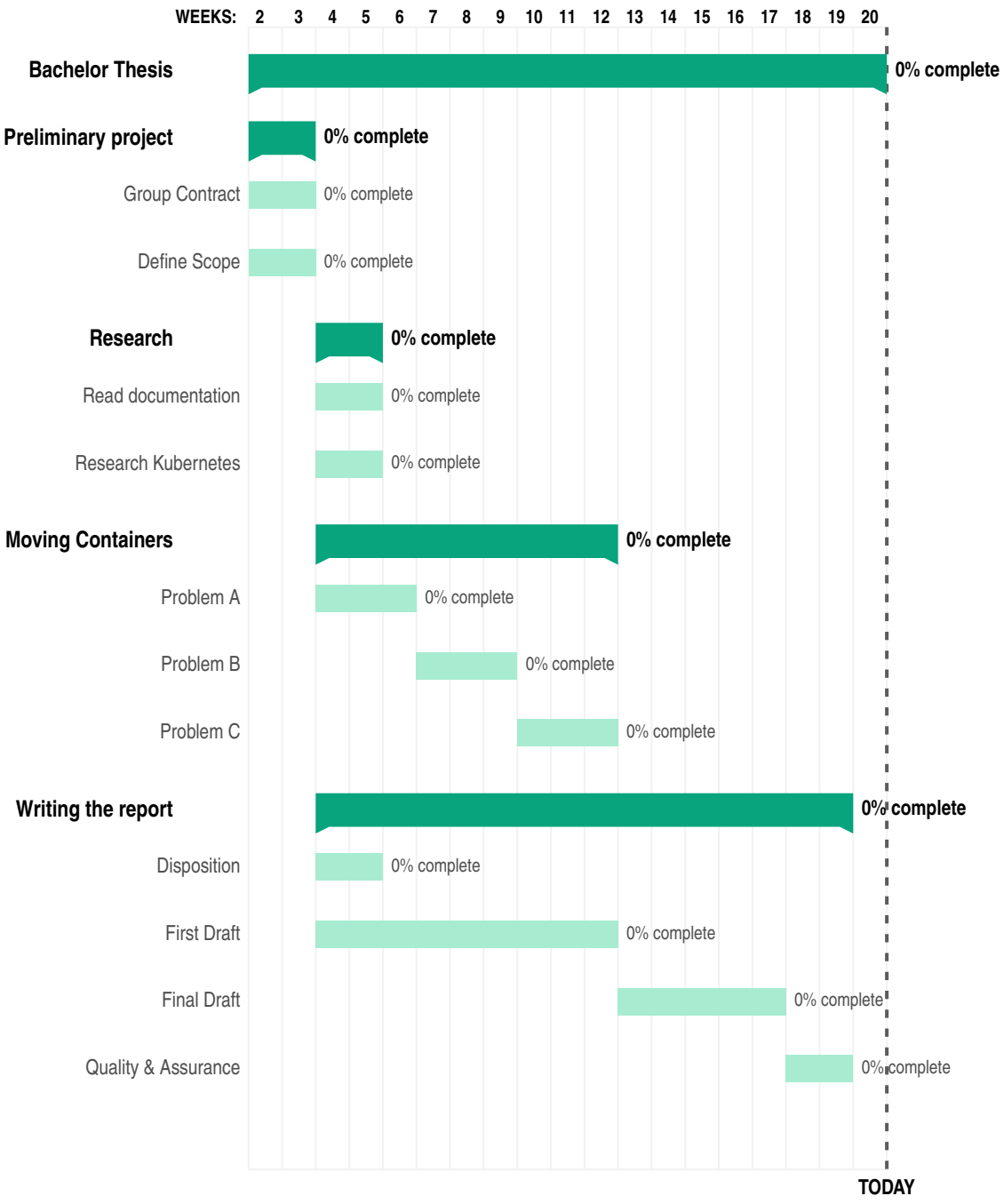
The Gantt-chart presented below compares our original plan created in the initial weeks of the project with the final chart that we updated regularly to reflect our progress. The red blocks on the chart indicate the time that we spent beyond our initial schedule. The chart clearly shows large deviations from our initial plan, mainly in the fields of *Research* and *Technical Work*. At the start of the project, we did not know which problems we would meet in the technical work, and only put problem A, B and C. This is the main reason for the large deviation here. The individual work processes got longer by the following percentages:

- Research: 150% increase
    - Read up on other technologies: Not part of the original chart
- Technical Work: 45% increase
    - Kubernetes Basics: 33% increase
    - Helm chart: 33% increase
    - Connecting the database: Not part of the original chart
    - Migrate data: Not part of the original chart
- Writing the report: 22% increase
    - Final Draft: 25% increase
    - Quality & Assurance: 33% increase

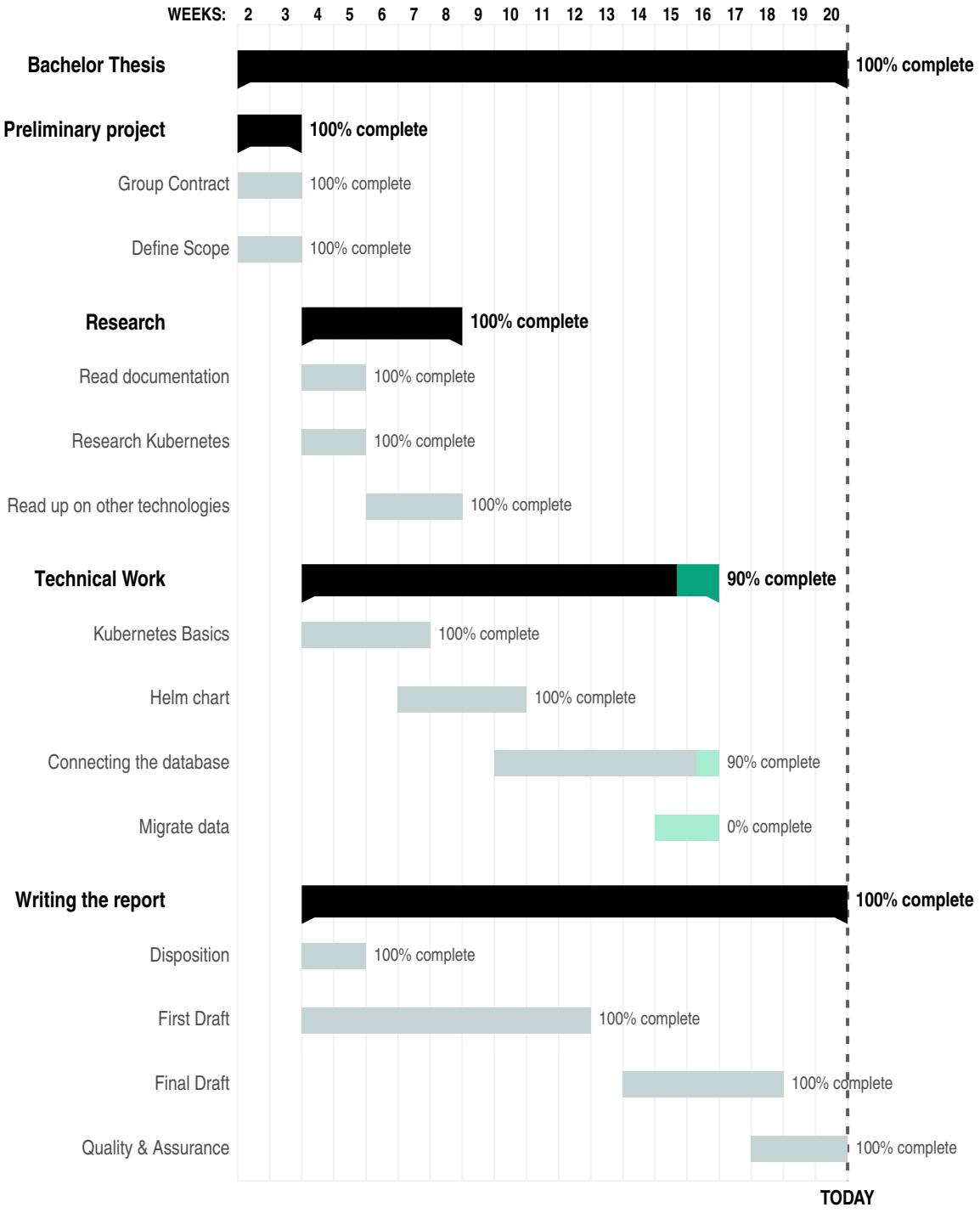
6.4.1 Differences in Gantt-charts



6.4.2 Original Gantt-chart



6.4.3 Final Gantt-chart



Chapter 7

Conclusion

7.1 Feedback From Client

The following feedback was received from Håvard Elnan, Senior System engineer at Norsk helsenett (NHN) on the 11th of May 2023.

About the project in general:

> *Providing an assignment in an emerging technology like Kubernetes is a challenge, as it imposes new and radical approaches to computing. Knowing this we provided a flexible problem definition that we intended to scope down during the project planning. In cooperation with the team, we came up with an assignment that was ambitious but still doable within the timeframe given. Cooperation with the team has been outstanding. Their ability to acquire knowledge on complex issues is also impressive, and questions asked is well considered and thought of. The result of the assignment is great both technically and theoretically.*

About starting with daily standups:

> *Moving in a terrain that is unknown is a difficult task and it's easy to get lost or fall into the pit of despair. By moving to daily standups, I think we shortened the time need to explore dead end paths and trying to climb out of a sinking hole.*

7.2 Conclusion

Based on our literature review and practical work with Azure, Kubernetes, ArgoCD, and MongoDB, we were able to develop a hypothetical migration model that can potentially work as a Kubernetes-based cloud solution. Our security analysis of migration to public cloud revealed that a hybrid cloud solution is necessary to ensure the confidentiality of the data managed by Norsk helsenett, as well as the recommendations and restrictions from the European Data Protection Board and General Data Protection Regulation. We concluded that sensitive data of level three, *Protected*, and four, *Heavily Protected*, should be stored on-prem while level one, *Open*, and two, *Internal*, data can be stored in the public cloud, allowing for the maximum possible migration of data to the public cloud.

Despite the limitations and challenges encountered during the thesis project, the group remains optimistic that our findings and proposed hybrid cloud solution can be of some value to Norsk helsenett. We hope that the theoretical model and practical approach we have presented can serve as a basis for future development and improvement of Norsk helsenett's cloud infrastructure.

7.3 Further Work

The practical part of the project requires continued troubleshooting to identify the underlying cause of why the MongoDB clusters will not add each other to the replication set. Once the cause of this issue is found, the next step should be to implement a solution and finish the migration setup as intended.

As of today, the chart only works on a Azure Kubernetes Service environment because it uses specific custom resource definitions that are specific for the Azure environment. Further work will be to make the chart available for different types of cloud options, like Tanzu or Amazon Web Services. This can be done by making boolean statements for the different custom resource definitions necessary that will be changed based on which type of Kubernetes cluster that will be deployed.

One of the goals we did not have the time to look at was the cost efficiency of moving data to a public cloud, but this would both be interesting and useful. A cost efficiency analysis would include finding out if hosting data on a public cloud instead of a private cloud would actually save any money, and if so, what would be the tipping point where a public cloud would save money. This would also include an analysis of NHN's current private cloud and see if it would be cost-effective to cut some of their cloud infrastructure when moving data to the public cloud.

In regards to the security aspect of the thesis, there is much more information that could be gathered about NHN. Conducting interviews with NHN employees and experts in the field would provide direct insight into the company's security culture and yield quantitative data to analyze. This approach would provide a more comprehensive understanding of the security practices within NHN and help

identify areas for improvement. For instance, safety measures that would reduce the risk to an acceptable level for both the third (Protected) and fourth (Heavily Protected) categories of sensitive data.

There are several potential areas for future research in the field of data privacy and security at NHN. With the increasing adoption of cloud storage by companies to store sensitive data, it is expected that new technologies, methods, or standards will emerge with great potential to improve our research and findings. As such, it is crucial to remain vigilant and keep up-to-date with the latest developments in the field to ensure the continued effectiveness and relevance of our research.

Bibliography

[1] Kubernetes. 'Overview.' (2023), [Online]. Available: https://kubernetes.io/docs/concepts/overview/ (visited on 23/02/2023).

[2] C. Strömholm, 'Judgement of the court: Schrems ii,' 2022. [Online]. Available: https://curia.europa.eu/juris/document/document_print.jsf;jsessionid=13C807006C7EB0F367E4423D8D1EADFF?pageIndex=0&docid=228677&doclang=EN&text=&cid=2035620 (visited on 13/03/2023).

[3] Microsoft. 'Imt software wiki - latex.' (2023), [Online]. Available: https://learn.microsoft.com/en-us/azure/aks/supported-kubernetes-versions?tabs=azure-cli (visited on 20/02/2023).

[4] Wikipedia. 'Kubernetes.' (2023), [Online]. Available: https://en.wikipedia.org/wiki/Kubernetes (visited on 09/03/2023).

[5] sokogskriv. 'Imrad-modellen.' (2023), [Online]. Available: https://www.sokogskriv.no/skriving/imrad-modellen.html (visited on 16/05/2023).

[6] S. for faglig kommunikasjon (SEKOM) and N. Universitetsbiblioteket. (n.d), [Online]. Available: https://i.ntnu.no/academic-writing/imrad-structure (visited on 07/03/2023).

[7] Overleaf. 'Tikz package.' (n.d), [Online]. Available: https://www.overleaf.com/learn/latex/TikZ_package (visited on 12/05/2023).

[8] W. Skala. 'Drawing gantt charts in latex with tikz.' (2018), [Online]. Available: https://ctan.uib.no/graphics/pgf/contrib/pgfgantt/pgfgantt.pdf (visited on 28/03/2023).

[9] T. U. of Edinburgh. 'Literature review.' (2022), [Online]. Available: https://www.ed.ac.uk/institute-academic-development/study-hub/learning-resources/literature-review (visited on 16/05/2023).

[10] Kubernetes. 'Learn kubernetes basics.' (2023), [Online]. Available: https://kubernetes.io/docs/tutorials/kubernetes-basics/ (visited on 29/03/2023).

[11] C. Rosen. 'Docker swarm vs. kubernetes: A comparison.' (2022), [Online]. Available: https://www.ibm.com/cloud/blog/docker-swarm-vs-kubernetes-a-comparison (visited on 13/03/2023).

[12] T. A. Limoncelli. 'Gitops: A path to more self-service it.' (2018), [Online]. Available: https://queue.acm.org/detail.cfm?id=3237207 (visited on 12/05/2023).

[13] ArgoCD. 'What is argo cd?' (n.d), [Online]. Available: https://argo-cd.readthedocs.io/en/stable/ (visited on 13/03/2023).

[14] T. Britten. 'What is helm?' (n.d), [Online]. Available: https://tanzu.vmware.com/developer/guides/helm-what-is/ (visited on 20/03/2023).

[15] Helm. 'Charts.' (n.d), [Online]. Available: https://helm.sh/docs/topics/charts/ (visited on 20/03/2023).

[16] R. C. Godoy. 'What is bitnami?' (n.d), [Online]. Available: https://tanzu.vmware.com/developer/guides/what-is-bitnami/ (visited on 21/03/2023).

[17] MongoDB. 'Why use mongodb and when to use it?' (n.d), [Online]. Available: https://www.mongodb.com/why-use-mongodb (visited on 12/05/2023).

[18] K. Morris, *Infrastructure as Code: Managing server in the cloud*. O'Reilly Media Inc, 2016, pp. 282–288.

[19] Cloudflare. 'What is hybrid cloud? | hybrid cloud definition.' (n.d), [Online]. Available: https://www.cloudflare.com/learning/cloud/what-is-hybrid-cloud/ (visited on 19/05/2023).

[20] Microsoft. 'What are public, private, and hybrid clouds?' (2023), [Online]. Available: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-private-public-hybrid-clouds/ (visited on 19/05/2023).

[21] A. Danielkievich. 'Top 10 data migration challenges and ways to overcome them.' (2023), [Online]. Available: https://forbytes.com/blog/common-data-migration-challenges/ (visited on 19/05/2023).

[22] K. Coco-Stotts. 'The top 5 threats to your it infrastructure.' (2020), [Online]. Available: https://jumpcloud.com/blog/five-threats-infrastructure (visited on 19/05/2023).

[23] C. P. S. Technologies. 'Top 15 cloud security issues, threats and concerns.' (2021), [Online]. Available: https://www.checkpoint.com/cyber-hub/cloud-security/what-is-cloud-security/top-cloud-security-issues-threats-and-concerns/ (visited on 19/05/2023).

[24] U. D. of Commerce. 'Privacy shield program overview.' (n.d), [Online]. Available: https://www.privacyshield.gov/Program-Overview (visited on 12/05/2023).

[25] O. of the Director of National Intelligence. 'Section 702 overview.' (n.d), [Online]. Available: https://www.dni.gov/files/icotr/Section702-Basics-Infographic.pdf (visited on 27/04/2023).

[26] E. D. P. Board, 'Recommendations,' 2021. [Online]. Available: https://edpb.europa.eu/system/files/2021-06/edpb_recommendations_202001vo.2.0_supplementarymeasurestransferstools_en.pdf (visited on 12/04/2023).

[27] E. Parliament and C. of the European Union, 'General data protection regulation,' 2016. [Online]. Available: https://gdpr-info.eu/ (visited on 12/05/2023).

[28] D. Silverberg and W. Smale, 'Home working increases cyber-security fears,' 2021. [Online]. Available: https://www.bbc.com/news/business-55824139 (visited on 12/05/2023).

[29] R. Cook, 'GitOps, an elegant tool for hybrid cloud kubernetes,' Presentation at USENIX LISA19, 2019. [Online]. Available: https://www.usenix.org/conference/lisa19/presentation/cook.

[30] Usenix. 'About usenix.' (n.d), [Online]. Available: https://www.usenix.org/about (visited on 27/03/2023).

[31] L. E. Pedersen. 'Openstack at ntnu.' (2023), [Online]. Available: https://www.ntnu.no/wiki/display/skyhigh (visited on 16/05/2023).

[32] A. CD. 'Installation.' (n.d), [Online]. Available: https://argo-cd.readthedocs.io/en/stable/operator-manual/installation/#supported-versions (visited on 20/05/2023).

[33] Openstack. 'Magnum compatibility matrix.' (2021), [Online]. Available: https://wiki.openstack.org/wiki/Magnum#Compatibility_Matrix (visited on 28/03/2023).

[34] MongoDB. 'Replication.' (n.d), [Online]. Available: https://www.mongodb.com/docs/manual/replication/ (visited on 11/05/2023).

[35] Microsoft. 'B-series burstable virtual machine sizes.' (2022), [Online]. Available: https://learn.microsoft.com/nb-no/azure/virtual-machines/sizes-b-series-burstable (visited on 10/05/2023).

[36] Microsoft. 'Network concepts for applications in azure kubernetes service (aks).' (2023), [Online]. Available: https://learn.microsoft.com/nb-no/azure/aks/concepts-network#kubenet-basic-networking (visited on 10/05/2023).

[37] A. Borucki. 'Deploying mongodb across multiple kubernetes clusters with mongodbmulti.' (2023), [Online]. Available: https://www.mongodb.com/developer/products/connectors/deploying-across-multiple-kubernetes-clusters/ (visited on 08/05/2023).

[38] BestVPN. 'Internet privacy index (2023).' (2023), [Online]. Available: https://bestvpn.org/privacy-index/ (visited on 20/04/2023).

[39] B. Cozens, 'Cloud data breaches: 4 biggest threats to cloud storage security,' 2022. [Online]. Available: https://www.malwarebytes.com/blog/business/2022/06/cloud-data-breaches-4-biggest-threats-to-cloud-storage-security (visited on 20/04/2023).

[40] E. Parliament and C. of the European Union, 'Gdpr: Art.49: Derogations for specific situations,' 2016. [Online]. Available: https://gdpr-info.eu/art-49-gdpr/ (visited on 26/04/2023).

[41] Azure. 'Azure geographies.' (n.d), [Online]. Available: https://azure.microsoft.com/en-us/explore/global-infrastructure/geographies/#overview (visited on 21/04/2023).

[42] E. D. P. Board, 'Recommendations,' 2021, pp. 3–5. [Online]. Available: https://edpb.europa.eu/system/files/2021-06/edpb_recommendations_202001vo.2.0_supplementarymeasurestransferstools_en.pdf (visited on 12/04/2023).

[43] E. Comission. 'Standard contractual clauses (scc).' (2021), [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection/international-dimension-data-protection/standard-contractual-clauses-scc_en (visited on 19/04/2023).

[44] E. Comission. 'Data governance act explained.' (2022), [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/data-governance-act-explained (visited on 12/04/2023).

[45] C. R. Service, 'U.s.-eu trans-atlantic data privacy framework,' 2022. [Online]. Available: https://crsreports.congress.gov/product/pdf/IF/IF11613 (visited on 12/04/2023).

[46] E. Comission. 'European data governance act.' (2022), [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/data-governance-act (visited on 25/04/2023).

[47] E. D. P. Board, 'Opinion 5/2023 on the european commission draft implementing decision on the adequate protection of personal data under the eu-us data privacy framework,' 2023, p. 6. [Online]. Available: https://edpb.europa.eu/our-work-tools/our-documents/opinion-art-70/opinion-52023-european-commission-draft-implementing_en (visited on 25/04/2023).

[48] H. Section, 'Matrix for risk assessments at ntnu,' 2010. [Online]. Available: https://i.ntnu.no/documents/1306938287/1306984199/Matrix+risk+assessments+-+eng.pdf/0d037956-7ea5-49db-94c3-512dc2ffdaff?t=1443197424892&status=0 (visited on 26/04/2023).

[49] Thales, '2022 thales data threat report: Navigating data security in an era of hybrid work, ransomware and accelerated cloud transformation,' 2022. [Online]. Available: https://cpl.thalesgroup.com/en-gb/euro-data-threat-report#download-popup (visited on 26/04/2023).

[50] C. S. Aliance, 'Top threats to cloud computing: Deep dive,' 2018. [Online]. Available: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-deep-dive/ (visited on 27/04/2023).

[51] Microsoft. 'Build in the cloud with an azure free account.' (n.d), [Online]. Available: https://azure.microsoft.com/en-us/free/ (visited on 08/05/2023).

[52] Microsoft. 'Build in the cloud free with azure for students.' (n.d), [Online]. Available: https://azure.microsoft.com/en-us/free/students/ (visited on 08/05/2023).

[53] H. Kniberg and M. Skarin, *Kanban and Scrum making the best of both*. C4Media Inc., 2010, pp. 4–5.

Appendix A

Preliminary project

Objectives and framework

Background

Norsk helsenett (NHN) is a IT company that deliver digital infrastructure and services for the Norwegian health sector. They are responsible for developing, managing, and operating various digital infrastructure and national e-health services such as helsenorge.no and the core health journal, all with the aim of improving the overall efficiency of the healthcare system.

NHN's private cloud is a platform that is built upon software-define infrastructure. It makes use of VMware Cloud foundation (VCF) and vSphere with Tanzu, so it is able to deliver both traditional and modern applications. The NHN private cloud also deliver a Kubernetes-platform where container services can run and internal customers/users can manage the their code and distribute their own services. As of today the Kubernetes-platform only runs on the NHN private cloud. Therefore NHN want to look at the possibility of running containers from the platform in a public cloud.

Main goals

Result goals

1. Designing a Kubernetes system that enables the movement of containers between private and public cloud environments.
2. Assessing to what degree a public cloud service provider meets the security standards set by NHN.

Effect goals

1. Enhancing the versatility of container hosting options and the ability to

choose the location where they are hosted.
2. Improve the overall performance, scalability and cost-efficiency of the NHN infrastructure while meeting the given security requirements.

Learning goals

1. Gain proficiency with Kubernetes.
2. Acquire experience in working collaboratively through being part of a team for an extended period of time.
3. Obtain practical experience through working with an actual client.

Framework

Time requirements

The minimum time requirement for the 18-week project period is 30 hours a week for each member. If more hours are needed, the time frame will be adjusted accordingly.

Language requirements

This thesis will exclusively use English as the writing language. The reasoning behind this decision is the prevalence of English-language resources, particularly in the fields of Information Technology and Kubernetes, making it more convenient when it comes to understanding technical phrases, terms, articles, and documentation.

Other frames

The thesis is built around Kubernetes as the main technology, so it is natural for the framework to be based on it. The project will use Tanzu Kubernetes, Openstack's Clusters and Azure Kubernetes Service.

Scope

Problem Area

The problem area for this thesis will focus on the implementation and integration of Kubernetes technologies, infrastructure, and cloud technologies. In addition to the implementation and integration aspects there will be a focus on the security aspect and how a public cloud service can measure up against NHN's required security.

Limitations

In the first meeting with NHN, the group was told that the project would be based on both Azure and Amazon Web Services. Later it was decided that it would be a waste time and resources to look at them both, considering they

work on similar bases. Therefore it was chosen that the group would focus on just Azure.

It was also decided that taking a look at the pricing model for Azure should not be a part of the bachelor thesis since it is not really relevant for the actual migration of the cluster. Therefore pricing will not be looked at unless the group has any extra time at the end of the process.

Problem description

The problem that we are trying to solve in this thesis, is to establish a Kubernetes platform on a public cloud that is able to work seamlessly with the private cloud infrastructure of NHN, allowing for the migration of container workloads between the two. In addition to this, we will evaluate the security aspect of the Kubernetes platform based on the security requirements from NHN.

Project organization

Responsibilities and roles

In order to have a resilient and structured task distribution, the group decided to have three main roles with different yet critically important responsibilities.

- **Leader:** The leader represents the group and is responsible for coordination of the group and managing internal conflicts.
- **Point of contact:** The point of contact is group's link between them and both the client, and the supervisor. The main responsibility of this role is to manage all the e-mail corresponding and keep the group up to date with necessary info obtained from the third parties.
- **Secretary:** The secretary is the group's writing hand having the responsibility for documenting all the meetings in a structured and clear way, so that the absent members can still benefit from the missed meeting. This role is also responsible for formatting of the main project document in LaTeX.

The roles are distributed between all of the three group members as following:

- **Leader:** Jonas Dale Fredriksen
- **Point of contact:** Sigurd Ose Dybdahl
- **Secretary:** Wiktor Miklaszewicz

Routines and group rules

Routines are an essential part of this group project, as they help to ensure that everyone is on the same page and the data stays intact. The contract is intended

to outline the expectations and responsibilities of each group member as the group work together to complete the project.

For the routines and group rules check the Group Contract (appendix B).

Planning, follow-up and reporting

Main division of the project

In order to work efficiently the group needs a structure to follow. One way to ensure efficiency and productivity is to use an already existing and well tested framework. A framework is a method of structuring daily tasks in a way that maximizes efficiency and gives a clear overview over the different stages of work progress.

The group decided to use a framework called Kanban. Kanban is a visual method for managing workflow that helps teams stay organized and focused. The key to using Kanban is to visualize the workflow by splitting the work into pieces, writing each item on a card, and putting the cards on a wall. This makes it easy to see what needs to be done and where each task is in the workflow. To further help with organization, Kanban uses named columns to illustrate where each item is in the workflow [53].

In this case the group decided to use a website named Trello.com to create a Kanban board and included the following columns:

TODO:
: Planned tasks that have not been assigned or taken by any member.

Ongoing:
: Tasks being worked on by a member.

Queue
: Tasks that are partially done and need some other task in order to get finished.

Check-list:
: Tasks that are finished need to be checked by all members in order to be considered as "done".

Done:
: Tasks that have been finished and checked by all members.

This structure allows the group to see at a glance which tasks are in progress, which are done, and which are blocked or stalled.

We selected Kanban as the foundation of our project, because some of the group members have had successful experiences with it in the past. Although Scrum was considered, it was ultimately deemed inappropriate for our thesis as the method primarily focuses on product development.

Plan for status meetings and decision-making process

Meetings

To maintain a consistent rate of progress, the group holds weekly meetings with the supervisor, the client, and the internal team members. The supervisor has extensive experience leading bachelor projects and is able to provide regular feedback on the progress and suggestions for improvement. This is crucial to ensure that the group does not fall behind and is able to deliver a high quality product in the given time.

The group's basic workweek consists of four different status meeting types:

- **Basic workday:**
  - Basic workday consist of working with previously planned tasks either digitally of physically.
  - Monday to Friday, 9:00 – 16:00(or earlier if the day's goals are achieved)
- **Meeting with the supervisor:**
  - Meeting with the supervisor consists of weekly guidance and updates of the group's progress.
  - Wednesday, 9:00 – 9:30
- **Meeting with the client:**
  - Meeting with the client consists of regular professional and technical guidance, as well as setting new goals if those set before were completed.
  - Thursday, 13:00 - 14:00
- **Week summary meeting:**
  - Week summary meeting consists of summing up the week's work and setting goals for the next week.
  - Friday, 14:00 – 14:30(or earlier if the day's goals are achieved)

Decision-making

Decision-making is a crucial aspect of any group project. It involves the process of identifying and choosing a course of action from among several alternatives. In order for the group project to be successful, it is important that all members of the group agree on the decisions that are made. This requires effective communication, problem solving, and collaboration. By working together, the group can ensure that the best possible decision is made for the project as a whole.

To achieve this, the group has decided to adopt a voting system as a way to make decisions. This means that the option chosen by the majority of group members

will be the one that is implemented. As the group only has three members, there is no possibility for a tie, thus not needing to include a third party voter.

The main goal is for all group members to agree on the decision, but if this is not possible, the voting system will be enforced. Only project-related conflicts will be subject to voting and any internal personal conflicts should be addressed according to guidelines in the Group contract (Appendix B).

Organization of quality of control

Documentation, standards, configuration and tools

Toggl:
The group will use track.toggl.com to track how much time is useed every day/week/month on the project, as well as categorizing the time used.

Trello:
Trello.com will be used as a Kanban board for the group. For more information about Trello and Kanban, see (Appendix A).

Overleaf:
The group will use a cloud-based LaTeX editor called Overleaf to write their thesis.

Slack:
The messaging platform Slack will be used to communicate with the client (NHN). In addition to getting in contact with our client, this can also be used to communicate with other people in the business if needed.

Teams:
The communication platform Teams will be used for remote meetings with NHN and correspondence with the supervisor outside of the weekly meeting.

Kubernetes:
Standards for Kubernetes is not known at the moment as the group does not have access to the documentation on the Kubernetes cluster that is already running at NHN.

AI tools:
The AI tools will be utilized as a learning aid during the initial stages of the project. Additionally, they will be used for creating code skeletons and detecting syntax errors. No sensitive information will be given to the AI. At this time, the specific AI tools that will be utilized have not been determined.

Plan for inspection and testing

The project centers around the Kubernetes platform. This platform consists of clusters which are the primary focus and will require thorough inspection and

Degree of probability	Probability description	Frequency interval(P)
Extremely Likely	More then eight times in the project period	$8/140 < P$
Very likely	Two to eighth times in the project period	$2/140 \leq P \leq 8/140$
Somewhat likely	One to two times in the project period	$1/140 \leq P < 2/140$
Not very likely	Less then once in the project period	$P < 1/140$

Table A.1: Probability description table

testing. The client currently primarily utilizes Helm lint and password detection in CI-pipelines for testing.

Helm lint checks packages with configuration files for issues such as syntax errors, incorrect usage of Kubernetes resources, and security vulnerabilities in the dependencies. It helps to ensure that the packages are well-formed and adhere to best practices.

Password detection is a process of identifying and flagging any plaintext passwords or other sensitive information that may be present in a package or its associated resource definitions.

They also have a developer working on a container that can assess the status of a cluster and possibly provide a demonstration for us.

We will utilize the same tools and techniques that our clients already have experience with, as they are experts in those methods and can assist us if needed.

Risk Analysis on a project level

The tables in this section are based on the Norwegian University of Science and Technology's risk matrix, but have been modified to suit our specific needs. The probability intervals for the frequency are determined by the number of days in the project, with a maximum of 140 days.

Probability description

See Figure A.1.

Consequence description

See Figure A.2.

Risk matrix

See Figure A.3.

Degree of consequence	Consequence description
Extremely consequence	The group is not able to deliver a satisfying final project
High consequence	The group is able to deliver the most important parts of the final project
Some consequence	The group is able to deliver a partially satisfying final project
No consequence	The group is able to deliver a satisfying final project

Table A.2: Consequence description table

Extreme likely				
Very Likely	1,7			9
Somewhat likely			8	4,5
Not very likely	3	6	2	10
	No consequence	**Some consequence**	**High consequence**	**Extreme consequence**

Table A.3: Risk matrix table

1. Not serious sickness:
The member that is sick will stay at home and contribute as much as they are able to digitally.
Mitigation: Staying at home if you are sick so no one else gets infected.
Consequence: 1
Probability: 3
Risk Score: 3

2. Serious sickness/injury:
Group will take over the responsibilities until the member is able to work again. If possible, the sick/injured member will do some work.
Mitigation: Stay away from dangerous activities. If you are sick, stay home.
Consequence: 3
Probability: 1
Risk Score: 3

3. Covid lockdown:
The group will have to start implement digital meetings and more individual work. It will be important to plan and manage the time and workload well.
Mitigation: Make sure you have access to all your files so you can work from home.
Consequence: 1
Probability: 1
Risk Score: 1

4. Client not able to give the needed assets and guidance, or there are communication problems:
The group will have to contact the supervisor and come up with an optimal solution. This will most likely result in continuing the project with a smaller scope or redoing the entire assignment.
Mitigation: Clear communication with the client and supervisor will be important to make sure the group has everything they need.
Consequence: 4
Probability: 2
Risk Score: 8

5. Data loss:
Try to restore from a backup. If this is not possible, the data that was lost will have to be redone.
Mitigation: Everyone will be required to take regular backups. There will also be a git repository that overleaf will automatically push to. The group members will pull from this project every couple days to make sure they have a recent backup on their own computer.
Consequence: 4
Probability: 2
Risk Score: 8

6. Internal conflict between the group members:
Contact the supervisor and find a solution.
Mitigation: Be a positive influence on each other.
Consequence: 2
Probability: 1
Risk Score: 2

7. All group rooms on campus are booked:
Work digitally or meet outside of the campus area.
Mitigation: Book rooms as far in the future as possible. Take 5 minutes each morning to book a room two weeks in advance.
Consequence: 1
Probability: 3
Risk Score: 3

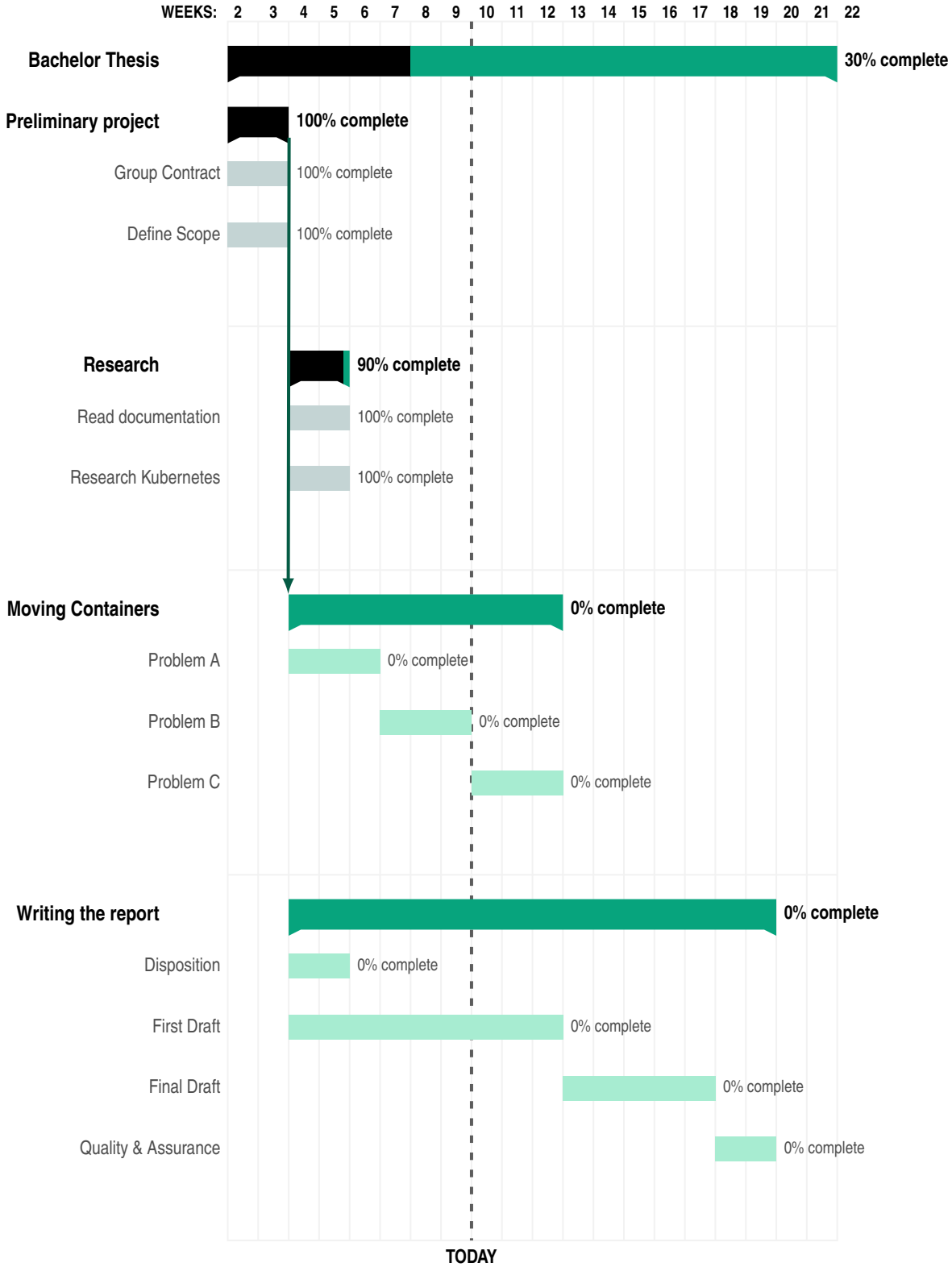
8. Part time job affecting the work quality:
The bachelor thesis must be prioritised. The affected members must talk with their boss and try to reduce the work amount. If this is not possible, the member will have to use more of their free time on the bachelor thesis.
Mitigation: Do not procrastinate your work. Get it done as soon as you are able to.
Consequence: 3
Probability: 2
Risk Score: 6

9. Scope mismanagement:
If the scope gets too big or too small, the supervisor will be contacted so the project gets back on the right track.
Mitigation: Good communication with the supervisor and the client to make sure the group is on the right track.
Consequence: 4
Probability: 3
Risk Score: 12

10. The group is not able to meet the required expectations:
The group will try to rewrite the assignment so they can deliver a satisfying product.
Mitigation: Spend the time you need to learn everything. If a week requires 40 hours of work, you need to put down 40 hours of work.
Consequence: 4
Probability: 1
Risk Score: 4

Project execution plan

Gantt-chart



Work Breakdown Structure

Milestones

Preliminary project done: To start on the main work the group need to have a preliminary project that provides them guidelines for the project as well as rules for the workflow.

Begin work on Kubernetes: After reading up on the documentation on NHN's private cloud platform the group can start working on how to move it securily to Microsoft Azure.

Moving containers between private Kubernetes cloud and Azure: The first part of the Kubernetes project is done. Next step is to make sure the containers are secure in transit and after they have been moved.

Kubernetes work done: All the implementation and development of the Kubernetes part of the project is done.

Start writing report: When the group has started working on the project, it is important to also start writing the report instead of waiting until the development is finished.

First draft delivered to Supervisor: The group will deliver the first draft of this thesis to our supervisor by March 31st.

Final draft delivered to Supervisor: After getting feedback on the first draft the group will use the feedback and revise the report to make it better. This final draft will be delivered May 1st.

Final report delivered: The final report will be delivered May 22nd.

Final Presentation done: A final presentation of our thesis will be presented in week 23, on either the 6th of June or the 7th.

Decisions done during our work:

Kanban Board: We decided to use a Kanban board using Trello to schedule what we are doing. Kanban was chosen because the group members had experience with it from other projects. For more information about our Kanban board see (Appendix A).

Toggle timechart: We decided to use Toggle to track our hours worked. This was done in a collaborative tool so everyone can see what the other members of the group has worked on lately and what they are currently working on. Tracking the hours in a collaborative excel chart was considered, but we

concluded that tracking three people for 20 weeks in an excel table would be a lot of data and it would look ugly.

Overleaf: We decided to write our thesis in the collaborative LaTeX editor Overleaf. This was done for easy collaboration, as well as previous negative experience in using collaborative Microsoft Word in a big project.

Only Azure, not Azure and AWS: We decided, together with our client NHN, that because of the way Kubernetes works similarly on Microsoft Azure and Amazon Web Services, we would get a better result when focusing deeper on only one of them instead of doing the same thing on two similar platforms. We will use Openstack to test our Kubernetes platform.

Appendix B

Contract

Rules

Attendance

1. The group's goal is to meet physically every workday (Monday - Friday), but it is mandatory to meet at least three times a week to maintain a good pace and achieve the desired goals.
2. The following meetings are mandatory: supervisor meetings, client meetings, and the Friday summary meetings. In the case of absence the absent member is obligated to read the meeting notes and documentation in order to catch up with the rest of the group.

Behaviour

1. The group's members are obligated to not let their free time attitude influence the work quality. The thesis should be a number one priority and the members should be professional with their personal conflicts.
2. If there are any personal conflicts internal in the group, do not let it influence the work quality and try to solve it in the free time.

Deadlines

1. The members are obligated to alert the others if they are not able to maintain a deadline in good time before the deadline.
2. If one is not able to maintain a deadline twice and causes a serious delay the supervisor is warned and the exclusion is considered.

Sanctions

If any member of the group fails to adhere to the rules, the other members will assess the severity of the rule violation. If the violation is deemed to be serious

enough the supervisor will be involved to further determine the penalty.

Routines

The group has two main routines, one for meetings and other one for backups.

Writing reports

1. The referent will document every summary meeting, supervisor meeting and client meeting.
2. This will ensure that every group member is on the same page and will secure the continuity, and fluency of the project.

Backups

1. The group will create weekly backups of the main document and other appendixes. The backups will be created after every Friday summary meeting and will have to be properly named relatively of the progress.
2. This will ensure data's integrity and minimize a possible recovery process, if something gets corrupted.

Appendix C

Result of time tracking

The following Figure C.1 shows two graphs representing our workhours during this thesis.

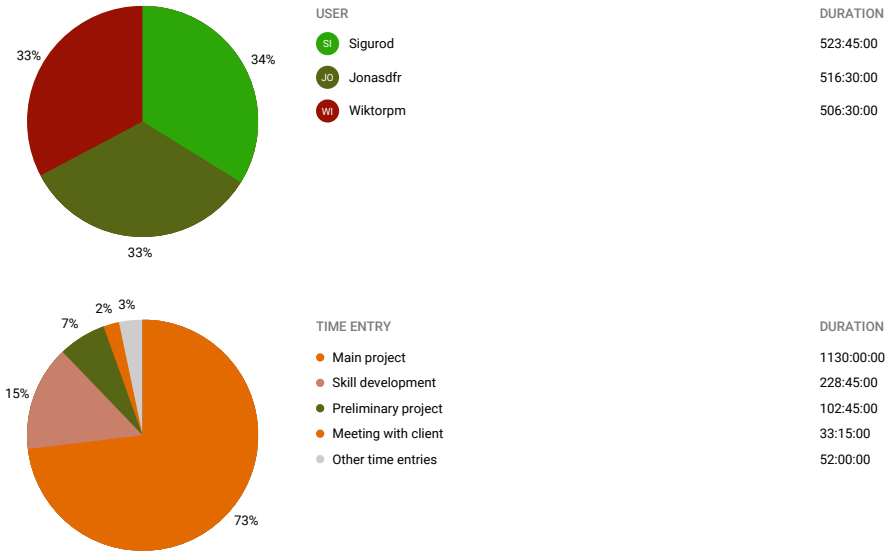


Figure C.1: The results of our time tracking in Toggle

Appendix D

Meetings with the supervisor

11. January Meeting

Date and time

11. January 2023, 9:00 - 9:30

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Get to know the supervisor
- Head start the project

Summary

We had a general discussion about the project and the resources needed in order to complete it. The main part of the project is a software called Kubernetes and we found out that the supervisor had a little bit of practical knowledge and few good resources like: Kelsey Hightower on Youtube.

We also got the main milestone dates for the project:

- Preliminary project before 31. January
- First draft of the main project before the Easter

The meeting ended with some suggestions about similar Bachelor theses that could aid us.

18. January Meeting

Date and time

18. January 2023, 9:00 - 9:30

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Get help with finishing the preliminary project

Summary

The meeting consisted mainly of us asking the supervisor questions about the structure of the preliminary project like:

- Correct English vocabulary
- Color palette
- Inspection and testing

After some discussion with the supervisor we came up with some conclusions:

- The report is to be written mainly in third person vies, to avoid writing a "story".
- The glossary and acronyms are only to be used in the main report, not the appendixes
- We are free to use any colors as we want, but try to not include the company's logo. It is a student project, not a company project.
- Although the project is not focused on "coding" the configuration in Helm is still considered as software development. Thus, still in need of inspecting and testing.

The meeting ended with a suggestion from the supervisor to move the meetings to 9:30 - 10:00 instead of 9:00 - 9:30.

25. January Meeting

Date and time

25. January 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Review the first draft of the preliminary project

Summary

The meeting consisted mainly of us and the supervisor discussing and reviewing the first draft of the preliminary project.

After some discussion with the supervisor we came up with some conclusions:

- The document has a lot of typos that has to be fixed.
- When a list only has one bullet point it is better to use "description list".
- The use of AI should be documented.
- All choices throughout the project should be justified and described.
- Use numeric bibliography.

The meeting ended with an overall positive feedback about the preliminary project.

01. Februar Meeting

Date and time

01. Februar 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Review the second draft of the preliminary project
- Quality check of the resources

Summary

The meeting consisted mainly of us and the supervisor discussing and reviewing the second draft of the preliminary project, and getting some high quality resources.

After some discussion with the supervisor we came up with some conclusions:

- The goals should describe the result we aim for not the activity needed to achieve the goal.
- Fill in more about backup (set up git repo)
- IoC book has some relevant models for migration (ex: blue green replacement)
- There was a bachelor thesis about Tanzu last year, that is worth checking out
- Usenix.org is THE BEST possible reference when it comes to making the thesis ""more scientific".

The meeting ended with conclusion that the preliminary project is finished.

08. February Meeting

Date and time

08. Februar 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Update the supervisor about our progress

Summary

The meeting consisted mainly of us updating the supervisor about the project's progress. We also discussed the proof-of-concept part of the project and came up with some conclusions:

- Create figures and models explaining our theories.
- Create a proof-of-concept dynamic application with mongodb that shows where it runs (replica, pod, cluster, location)

The meeting ended with conclusion that we are on right course.

15. Februar Meeting

Date and time

15. February 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Update the supervisor about our progress
- Try to get unstuck

Summary

The meeting consisted mainly of us updating the supervisor about the project's progress and that we feel stuck. After the conversation we concluded with the following:

- Focus more on Azure.
- Openstack is limiting us, because of it's legacy version of Kubernetes.
- Set up Helm charts that rolls out applications and databases.

The meeting ended with conclusion that we are stuck and have to consult the client.

22. February Meeting

Date and time

15. February 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Update the supervisor about our progress and finds

Summary

The meeting consisted mainly of us updating the supervisor about the project's progress and that we got unstuck by creating the proof of concept in Azure. After the conversation we concluded with the following:

- Inform NTNUs SkyHiGh that the Kubernetes version is outdated.
- Go more in depth on why the Kubernetes version is outdated.
- Contact Gaute in order to review the finds.
- Remember to take screenshots and create figures.

The meeting ended with conclusion that we are no longer in the pit of despair and there is a light in the tunnel.

01. March Meeting

Date and time

01. March 2023, 8:30 - 9:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Discuss project's structure

Summary

The meeting consisted mainly of us discussing thesis' structure. After the conversation we concluded with the following:

- Use IMRAD structure
- Remember to include a theory chapter after method

The meeting ended with conclusion that we should come up with the first structure draft before the next meeting.

08. March Meeting

Date and time

01. March 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Furthere discuss project's structure

Summary

The meeting consisted mainly of us further discussing thesis' structure, mainly the Background Theory. After the conversation we concluded with the following:

- Create a structure draft for the next meeting

15. March Meeting

Date and time

15. March 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Further discuss project's structure

Summary

The meeting consisted mainly of us further discussing thesis' structure, mainly the Method Theory.

22. March Meeting

Date and time

22. March 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Present structure draft
- Further discuss project's structure

Summary

The meeting consisted mainly of us further discussing thesis' structure. We also presented our structure draft and got a positive feedback about our progress rate.

29. March Meeting

Date and time

29. March 2023, 9:45 - 10:15

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Further discuss Method section

Summary

The meeting consisted mainly of us further discussing thesis' structure, mainly Method. The problem was that we were not entirely sure how to divide practical part of Kubernetes and security, as well as what to put in the actual Method. After the conversation we concluded with the following:

- Create a new section called Method and put it behind Practical Implementation.
- Divide the Kubernetes and security i two different sections
- Have a first draft ready for 1. April.

12. April Meeting

Date and time

12. April 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- General feedback about our progress and the 1. draft

Summary

The meeting consisted mainly of Erik giving us feedback about our progress and the first draft.

Feedback concluded with the following:

- Do not write with personal pronoun where it is not needed.
- The draft was satisfying and we are on a good path.
- We need to start thinking about what we will do if we cannot create a proof-of-concept.

19. April Meeting

Date and time

19. April 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- General feedback about bachelor grading

Summary

The meeting consisted mainly of Erik discussing with us about how bachelor theses are graded. He told us the main features that are graded in a bachelor thesis. Erik also asked if we could show him what we have written since last meeting.

Feedback concluded with the following:

- Check if our thesis is "scientific".

- Show our progress on every upcoming meeting.

26. April Meeting

Date and time

26. April 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Discussion about stopping the practical part.
- What comes next?

Summary

The meeting consisted mainly of us and Erik discussing if we should stop the practical part of the thesis, since we are not able to figure out how to fix our problem. And the deadline for the report is approaching very fast.

Feedback concluded with the following:

- The report if the most important part of the thesis, so it should be downprioritized.
- We can try to fix the proof-of-concept if there is some time left after finishing the report.
- Remember to show the knowledge gained through the project period

03. May Meeting

Date and time

03. May 2023, 9:30 - 10:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Discussion about result, goals and conclusion part.

Summary

The meeting consisted mainly of us and Erik discussing how we should form the result, since we have no quantitative results, rather a good conclusion. We also

showed our Gantt charts and the difference between one created in the preliminar project and on that actially reflecs our work.

Feedback concluded with the following:

- Explain detailed why and how some goals were not met.
- Add all Gantt chart together as appendix, but add one showing the difference between two in the actual report.
- Add working hours as appendix.

10. May Meeting

Date and time

10. May 2023, 13:00 - 14:00

Present

Erik Hjelmås, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Feedback from the final draft.

Summary

This was our last meeting with the supervisor. We went through his feedback on our final draft in detail. The overall report looks great, but still needs some fixes here and there.

Feedback concluded mainly with the following:

- Try to not use too many short acronyms, use preferably long ones.
- Every list, figure and listing HAS to be refered in the text.
- Remove any "salespropaganda".
- It is not necessary to start every section with an introduction, just write directly about the topic.
- According to the supervisor, the report has no significant deficiencies.

Appendix E

Meetings with the client

12. January Meeting

Date and time

12. January 2023, 13:00 - 14:00

Present

Erik Hjelmås, Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Introduce the client and the supervisor to each other
- Narrow down the project scope
- Copyright privileges
- Regular meeting

Summary

The beginning of this meeting was meant to intoduce our supervisor Erik Hjelmås to our client Norsk helsenett (NHN)

We used the opportunity to narrow down and come up with a precise project scope. With the help of the NHN and our supervisor we managed to establish the main goal of the project:

- Establish a optimal platform in Microsoft Azure

And the thesis statements:

- What is necessary migrate over to Azure (AKS) from a private cloud in a most optimal way

- The main differences between Azure and the on-prem private cloud
- Security requirements related to migration from a private to a public cloud

We agreed to have the regular meetings with the client once a week in the beginning to get as much guidance as possible. The meeting will take place on Teams on Thursdays 13:00 - 14:00. Sigurd Ose Dybdal is responsible for arranging the meetings and inviting the participants.

The meeting concluded with a small to do list:

- Read about Azure and AKS
- Read the code documentation on Git
- Send the copyright papers to NHN HR
- Fix a quota on Openstack

19. January Meeting

Present

Håvard Elnan, Sigurd Ose Dybdal and Jonas Dale Fredriksen

Agenda

- Questions from our side
- What has happened since last week

Summary

We have been working on the preliminary project the last week and we just delivered it, so we are waiting on feedback on that before we can start working on the main project. We asked some questions about the points in the preliminary project that we were missing:

- The Kubernetes version that is being used is VMWare Tanzu 1.2.8.
- The testing being done is helm lint and a password detection i CI pipeline
- A container that will inspect the status of the other containers is being worked on.

In addition to this, we made a to do list for tomorrow, which only consists of one point:

- Send github handles so we can be added to the git repo

02. February Meeting

Meeting canceled.

09. February Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl and Wiktor Pawel Miklaszewicz

Agenda

- Demo by the client showing the deployment of the toolkit to ArgoCD.

Summary

The client showed of a live demo of the deployment of their toolkit to ArgoCD. We also got a quick lesson on how Kubernetes namespaces and contexts works in practice.

We agreed to try and deploy the toolkit on our owm after the meeting.

16. February Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Questions from our side
- What has happened since last week
- How do we get unstuck?

Summary

In the past week we felt stuck without any progress and therefore came to the client for some help. The main problem was the connection between Azure and Openstack. We could not find a solution for exposing an Openstack cluster to Azure. We also found out that the version of Kubernetes used on Openstack is outdated and with therefore not work with Azure.

After some discussion with the client, we concluded with:

- We will assume that the network connection is okay and simulate the whole process inside of Azure, in order to deliver a proof-of-concept.
- Since we are in the heavily practical part of the project the client agreed to have daily stand-up meetings with us at 14:00.
- We were also told to look closer on "Schrems 2" case.

23. February, 02., 09. and 16. March Meetings

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Questions from our side
- What has happened since last week

Summary

These three meetings were quite quick since we have had daily stand-ups, and got continuous help if something went wrong. The group is in a full on writing mode so there is actually not so much we have to discuss with the client.

13. April Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Questions from our side
- What has happened since before the Easter break
- Problem with MongoDB

Summary

This meeting served as a catch up after the Easter break. We presented what we have written on paper until now and then we discussed the problem with MongoDB.

After some discussion with the client, we concluded with:

- One of us will have a one-on-one session with the client next week in order to try and fix the MongodDB problem.

20. April Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Questions from our side
- What has happened since last week

- What happens now since we are not able to finish the proof-of-concept?

Summary

After the one-on-one session, together with the client we did not manage to find out a solution, but we have a suspicion that MongoDB requires that you purchase the Enterprise version in order for the replicas to work.

After some discussion with the client, we concluded with:

- We stance the work with proof-of-concept and focus on the report.
- If we are finished with the report and have some time remaining we will come back and try to create a working proof-of-concept.

27. April Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Questions from our side
- What has happened since last week

Summary

This was a short meeting since we did not have so much to talk about, mainly because we are just writing the report at the moment. But, the client suggested that if we have some time remaining, we could try to set up Redis database instead of MongoDB. This is not as dynamic as MongoDB, but should be enough to visualise the concept.

After some discussion with the client, we concluded with:

- We will no longer have daily stand-ups, since the technical phase is over.
- If there is time remaining before the deadline, try to create a proof-of-concept using Redis.

04. May Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Feedback about the whole project

Summary

This was a very short meeting since we did not have so much to talk about, mainly because we are just finishing the report at the moment.

We sent the almost final draft to the client in order to get a feedback from their perspective on the task and the general thesis.

After some discussion with the client, we concluded with:

- Håvard will come back with feedback when he is finished reading the report.

11. May Meeting

Present

Håvard Elnan, Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Agenda

- Feedback about the whole project

Summary

This was the last meeting with out client. Håvard presented his feedback about the overall project and its difficulty level. The feedback about our cooperation can be summed up with the following quote:

> *Cooperation with the team has been outstanding. Their ability to acquire knowledge on complex issues is also impressive, and questions asked is well considered and thought of.*

Appendix F

Meetings with the group

13. January Meeting

Date and time

13. January 2023, 14:00 - 14:30

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This week was used a preparation week.

We had the first official meetings with the supervisor and the client. Which both succeeded with plenty of resources to successfully get a head start on the project.

We got the Openstack quota, booked meeting rooms for the next week and the client is preparing the necessary documentation we need to read. We have also fixed all the necessary administrative tools like Toggl, Trello, Slack, Teams and Openstack document.

The group rules and routines are finished and we also have a first draft of the contract.

Plan for the next week

- Finish the preliminary project
    - Finish the contract
    - Translate titles to English
    - Write down a clear project goal and issues
    - Create the risk matrix
    - Rest of the preliminary project

- Start to read the documentation on code and Azure and AKS and Kubernetes

20. January Meeting

Date and time

20. January 2023, 14:00 - 14:30

Present

Sigurd Ose Dybdahl and Jonas Dale Fredriksen

Week summary

This week we finished the preliminary project and sent it to our supervisor to get feedback. We also booked meeting rooms for the two next weeks. In the meeting with our client we got the necessary information we were missing, and will hopefully be added to the github repo with documentation in the next couple days.

Plan for the next week

- Read about Kubernetes, AKS and Tanzu.
- Start to dabble with Kubernetes to understand the basics
- Fix the preliminary project based on the feedback we get.

27. January Meeting

Date and time

27. January 2023, 14:00 - 14:30

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This week was used for skill development.

We started learning the basics of Openstack and Kubernetes, and tried to set up a K8s environment in Openstack. It went well and the whole group managed to set up at least one functioning cluster with a simple HTML website.

We also got feedback from the first draft of the preliminary project and fixed all mentioned fails, and sent the second draft to the supervisor.

Plan for the next week

- Set up and scale up a k8s environment
- Read the documentation on Helm and try to set up a system using it

- Read documentation about migrating of containers.

03. February Meeting

Date and time

03. February 2023, 11:30 - 12:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This week was also used for skill development.

We have learned the basics of Kubernetes and Openstack. We have also been researching Helm, ArgoCD and Prometheus.

We have gained a good makro perspective on the whole project and started to lay out a plan on how the migration will be carried out.

Github repo has been established and we are still working on connecting the Overleaf document to automate backup process.

We also got feedback from the second draft of the preliminary project and fixed all mentioned fails, and sent the final draft to the supervisor.

Plan for the next week

- Prepare to actually migrate containers
- Build a test environment
- Build a test cluster
- Try to migrate containers

10. February Meeting

Date and time

10. February 2023, 11:30 - 12:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This week was used for preparation to actually start trying to migrate containers.

We have learned the basics of Kubernetes, Openstack and ArgoCD.

Got a demo from client to get a better view of their Kubernetes setup.

We also started with Azure, but we use the trial version since the client have not provided us with license keys.

Plan for the next week

- Hope for Azure licenses
- Start to develop figures
- Look at other reports (all read one unique bachelor)
- Check models in book

17. February Meeting

Date and time

17. February 2023, 13:00 - 13:30

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This week was a dark one. We ended up in the pit of despair.

Luckily the supervisor and the client helped us out of our misery and there was a light in the tunnel again.

We decided to only use Azure, because of Openstacks legacy Kubernetes version and limitations connected to networking problems.

Plan for the next week

- Hope for Azure licenses
- Start to develop figures
- Create a helm chart for application
- Kubernetes to kubernetes migration
- Look closer on the security risks connected to data migration

We decided to end the workday at 13:30, since we are still recovering from the pit of despair.

24. February Meeting

Date and time

24. February 2023, 13:30 - 14:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This week was a much better than the last one.

We are on very good track. The Helm charts are almost finished, all errors from our LaTeX report are fixed, and we have gathered a lot of good quality scientific documents about data migration security. We have started some work on setting up a mock database in MongoDB so we can migrate something more complex than a static nginx server.

We also ran home office the whole week, because of sickness, but we are hoping to get back to being physically on campus the next week.

Plan for the next week

- Hope for Azure licenses
- Finish the helm chart for application
- Finish the MongoDB database
- Kubernetes to kubernetes migration
- Create the report structure
- Start ti fill out what we can in the report

10. March Meeting

Date and time

10. March 2023, 13:30 - 14:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

We are on a good track. Part of the group have been filling oit the Background Theory section, while the other part is still trying to figure out the mongoDB.

Plan for the next week

- Hope for Azure licenses
- Finish the MongoDB database
- Start to fill out what we can in the report

17. March Meeting

Date and time

17. March 2023, 13:30 - 14:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

We have finally received the Azure licenses. Background Theory section is almost finished. We began the work on the Method, but we are struggling with what to put in Method and how to combine k8s and data privacy.

We still struggle with mongoDB.

Plan for the next week

- Finish the MongoDB database
- Ask the supervisor for help with the structure
- Come up with the Method structure

31. March Meeting

Date and time

31. March 2023, 13:30 - 14:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

We managed to create a satisfying structure combining both k8s and security, after the consultation with our supervisor. The Method section has also been created. The Azure licenses from NHN do not work.

The struggle with MongoDB is ongoing.

Plan for the next week

- Contact NHN and try to get the licenses to work
- Fix MongoDB
- Start filling out the Method section

14. April Meeting

Date and time

14. April 2023, 13:30 - 14:00

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

We started the week with catching up after the Easter break. We did not get any answers about the licenses yet. There is still problem with replicas in MongoDB, but we have to start writing about the results soon.

Plan for the next week

- Contact NHN again about the licenses
- Try to fix MongoDB
- Start writing about our goals

21. April Meeting

Date and time

21. April 2023, 14:00 - 14:30

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

The deadline is approaching pretty fast so we decided that the next week will be the last with practical work and if the problems are not fixed we will just present what we have without a proof-of-concept. The licenses stil do not work.

Plan for the next week

- Finish the Practical implementation
- Contact NHN again about the licenses
- Try to fix MongoDB

28. April Meeting

Date and time

28. April 2023, 14:00 - 14:30

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

We are now officially down prioritized the practical part. There is one tiny thing that we are not able to fix and the deadline for the report is coming at a dangerous paste. We are also almost finished with the Practical Implementation and plan to finish 95% of the report the next week.

Plan for the next week

- Finish Practical Implementation.
- Start and try to finish Results.
- Start and finish Further Work.
- Try to form a conclusion.

05. May Meeting

Date and time

05. May 2023, 14:00 - 14:30

Present

Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

The group is almost finished with the final draft. The only thing remaining is to read through the report before sending it to supervisor and client on May the 8th.

Plan for the next week

- Read through the final draft.

12. May Meeting

Date and time

12. May 2023, 14:00 - 14:30

Present

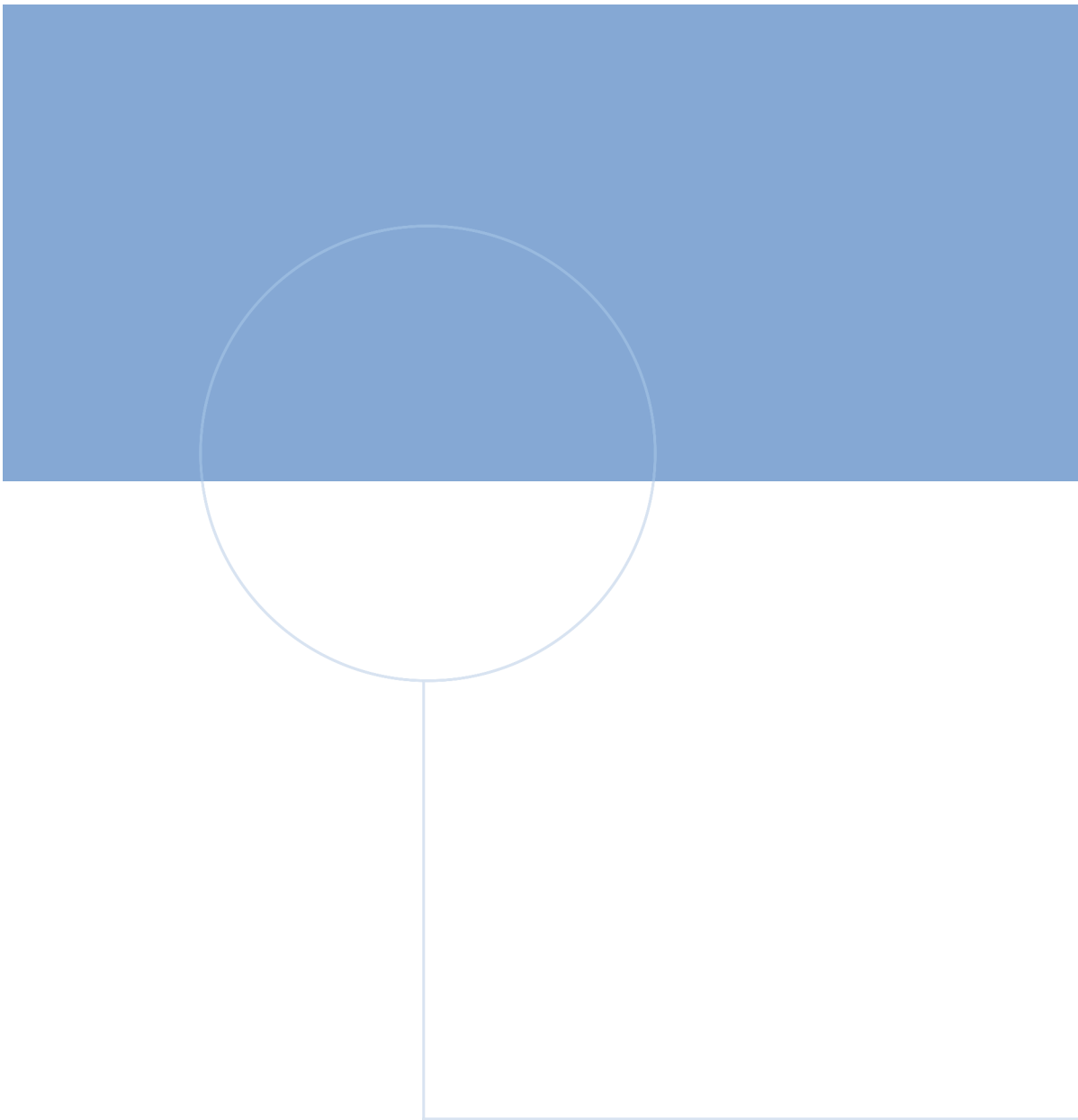
Sigurd Ose Dybdahl, Jonas Dale Fredriksen and Wiktor Pawel Miklaszewicz

Week summary

This is the last meeting with the group. We have received final feedback from both the supervisor and the client. We fixed most of the supervisor's comments and are almost ready to deliver the final thesis.

Plan for the next week

- Fix glossary and acronyms.
- Write a summary to have in Abstract.
- Read through the final project together and correct any mistakes on the way.



NTNU

Norwegian University of
Science and Technology