Defining and Measuring Efficiency gains through Infrastructure as Code Adoption

A MultiVocal Literature Review and
a case study on a company

Alexandru-Constantin Petrovics
Computer Science, CS-IT10, 2024-06

Master Thesis

STUDENT REPORT
AALBORG UNIVERSITY

This report is submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science from Aalborg University, Denmark.

AALBORG UNIVERSITY
STUDENT REPORT

Electronics and IT
Aalborg University
http://www.aau.dk

Title:
A study on Infrastructure as Code

Theme:
Scientific Theme

Project Period:
Spring Semester 2024

Project Group:
cs-24-sd-10-03

Participant(s):
Alexandru Constantin Petrovics

Supervisor(s):
Adam Alami

Copies: 1

Page Numbers: 83

Date of Completion:
June 10, 2024

Abstract:

The first part, a Multi-Vocal Literature Review (MLR), identified four key outcomes of IaC efficiency: Speed Increase and Cost Reduction, Scalability and Standardization, Security and Documentation, and Disaster Management. Twenty-two key metrics were identified, with the most frequently mentioned being Lead Time for Changes, Time to Restore Service, Change Failure Rate, Deployment Frequency, and Deployment Time Duration.
The second part, a case study at Jumia, focused on understanding the company's approach to IaC efficiency. Interviews highlighted critical metrics such as Infrastructure Code Coverage, Time to Restore Service, Number of Drifts, Deployment Frequency, Lead Time for Changes, and Change Failure Rate. These metrics closely aligned with those identified in the MLR, indicating a shared understanding of IaC efficiency between academia and industry. Moreso, the benefits and drawbacks of IaC experienced by Jumia also closely relate to the literature. The findings serve as a reminder that while there is a consensus between academia and practice, IaC efficiency evaluation should be tailored to the specific needs of the organization.

Contents

Preface

0.1 Acknowledgement

I extend my gratitude to all those who have contributed to the completion of this paper. Special thanks to my supervisor Adam Alami for his guidance and support throughout the research process. I also want to thank the individuals from Jumia for their cooperation and help with the interviews. This work is a culmination of collective effort and collaboration, and I am truly thankful for the encouragement and assistance received.

Aalborg University, June 10, 2024

Alexandru-Constantin Petrovics

<apetro21@student.aau.dk>

Chapter 1

Summary

Introduction In the dynamic landscape of modern software development, organizations face increasing challenges in delivering reliable, scalable, and bug-free software. DevOps has emerged as a pivotal methodology that integrates software development (Dev) and IT operations (Ops) to accelerate the software development lifecycle and ensure continuous delivery. A core practice within DevOps is Infrastructure as Code (IaC), which involves managing and provisioning computing infrastructure through machine-readable scripts rather than manual processes. IaC enables teams to apply software development practices such as version control, continuous integration, and automated testing to infrastructure management, improving deployment speed and quality, enhancing collaboration, and ensuring consistency and reproducibility.

Aim Despite the growing popularity of DevOps and IaC, academic literature focusing on practical aspects of IaC is sparse. There is a significant gap in understanding how to measure efficiency improvements from IaC implementation. Most existing studies concentrate on DevOps as a whole, leaving a lack of clarity on specific metrics related to IaC efficiency. This thesis aims to bridge this gap by identifying how practitioners and literature sees this efficiency, and by identifying and prioritizing key metrics used by practitioners to evaluate IaC efficiency, thus providing actionable insights for both researchers and industry professionals.

Research Questions:

1. Multi-Vocal Literature Review:
    - How do software engineering practitioners define "efficiency" of IaC implementations?
    - What are the key metrics reported in the literature to evaluate the efficiency of IaC implementations?

2. Case Study:
   - How do software practitioners expect to evaluate the efficiency of IaC implementations?
   - How do the challenges and benefits associated with IaC implementation at Jumia compare to the broader literature?

Methodology To answer the research questions, the methodology of this thesis was two-fold, involving a Multi-Vocal Literature Review (MLR) and a case study. The MLR was conducted following the guidelines by Garoussi et al.[58],and included both academic and non-academic sources. The second part involved a case study at Jumia, an e-commerce company, and explores their implementation of IaC. Semi-structured interviews were conducted with practitioners to understand their perspectives on IaC efficiency, challenges, and benefits. This approach allowed for a detailed comparison between theoretical insights and real-world applications, providing a well-rounded view on efficiency in IaC.

Results:

Multi-Vocal Literature Review The Multi-Vocal Literature Review (MLR) identified four key outcomes of IaC efficiency: Speed Increase and Cost Reduction, Scalability and Standardization, Security and Documentation, and Disaster Management. Twenty-two key metrics were found, with the following five being most frequently mentioned: Lead Time for Changes, Time to Restore Service, Change Failure Rate, Deployment Frequency, and Deployment Time Duration. These metrics highlight the importance of rapid, reliable, and frequent software deployments.

Case Study The case study at Jumia revealed a primarily reactive approach to IaC, focusing on immediate infrastructure health and service availability. However, when asked about critical efficiency metrics, Jumia practitioners highlighted several that aligned closely with the MLR findings. These included Infrastructure Code Coverage, Time to Restore Service, Number of Drifts, Deployment Frequency, Lead Time for Changes, and Change Failure Rate. This alignment suggests a shared understanding of key IaC efficiency indicators between theoretical and industry practices. Jumia's primary focus on cost reduction and time savings also aligns with traditional definitions of software engineering efficiency. The case study also identified the challenges and benefits of adopting IaC. Challenges included significant time investment for migration and initial setup, learning curves with IaC tools and languages, and potential consequences from configuration mistakes. Meanwhile, benefits included improved standardization, automation, enhanced disaster recovery, and fostering a DevOps culture.

Chapter 2

Introduction

In today's fast-changing world, organizations,from startups to corporations, are facing more and more issues with keeping up with the delivery requirements of reliable, scalable, and bug-free software. As a consequence, during the last decade DevOps has become a rising practice in the Software development industry. DevOps is a methodology, comprised of a set of practices and tools that combines software development (Dev) and IT operations (Ops) with the aim of shortening the software development life cycle and providing continuous delivery of software. The methodology puts a high emphasis on cross-team communication and collaboration, and technology automation.

Infrastructure as Code (IaC), as defined by John Klein [23] is "a set of practices that use code (rather than manual commands) for setting up (virtual) machines and networks, installing packages, and configuring the environment for the application of interest. The infrastructure managed by this code includes both physical equipment ("bare metal") and virtual machines, containers, and software-defined networks.". IaC is an essential DevOps practice and a component of continuous delivery.

By treating infrastructure as code, teams can apply software development practices like version control, continuous integration, and automated testing to infrastructure management. This not only improves the speed and quality of deployments but also enhances collaboration by allowing developers and operations teams to work from a single source of truth. This allows for easier history management, higher traceability of changes, easier rollback in case of failures, and preserves immutability. Immutability happens when changes that are not recorded are done to the infrastructure, with environments changing in non-reproducible ways. Another characteristic and advantage of IaC is idempotence. This ensures that the final states of an environment will be the same, regardless of the starting state. This simplifies the infrastructure provisioning, while also reducing the chance of inconsistencies happening. [47]

2.1 Research Problem

Despite the increasing popularity of DevOps and IaC, there is a marked lack of white (academic) literature with a focus on a more practical area. Multiple studies conducted in this area made a remark that IaC needs more focus from academics([14], [30]). On the other hand, there has been a noticeable increase in the interest given to this topic, shown both by data from Google Trends, and by a large number of grey (non academic) and black literature. Besides, most literature is focused on DevOps as a whole. With this comes a lack of understanding about how to measure the efficiency improvements through the use of IaC. While there are numerous metrics available, it is unclear which ones are related to IaC, and prioritized by practitioners in the field. This gap in knowledge leads to suboptimal practices and missed opportunities for improvement. Figure 2.1 shows the Google Trend data related to the search term "Infrastructure as Code".

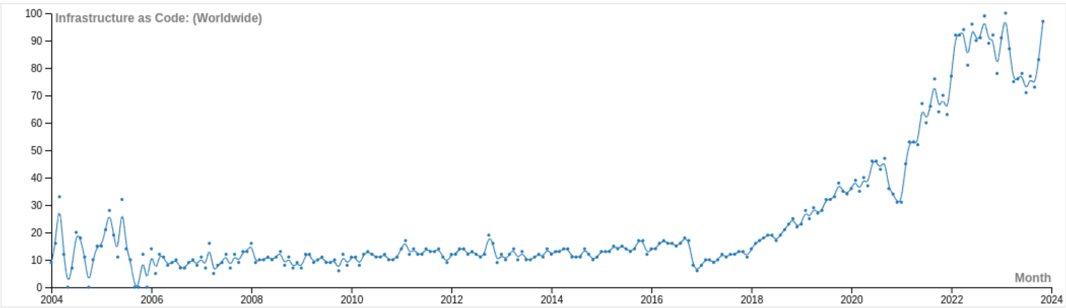

Figure 2.1: Google Trends for Topic 'Infrastructure as Code'

On the other hand, figure 2.2 shows the number of publications written around the topic "Infrastructure as Code" according to Dimensions, showing a yearly increase.

One thing to note: DevOps is a much more studied topic, for example in 2022 there were over 2500 papers published(according to Dimensions), more than 5 times the number of topics published in the same year on Infrastructure as Code.

2.2 Research Objectives

This thesis's objectives are split into two parts:

2.2.1 Part 1: Defining and Measuring Efficiency in IaC

The first part of the thesis aims to address the gap in knowledge on defining efficiency, and key metrics for measuring the efficiency improvements gained through the adoption of IaC by leveraging both white and grey literature.

The specific objectives of the first part are:

- To define efficiency in the context of IaC
- To identify commonly used metrics for measuring efficiency improvements in IaC
- To prioritize these metrics based on information received from DevOps practitioners

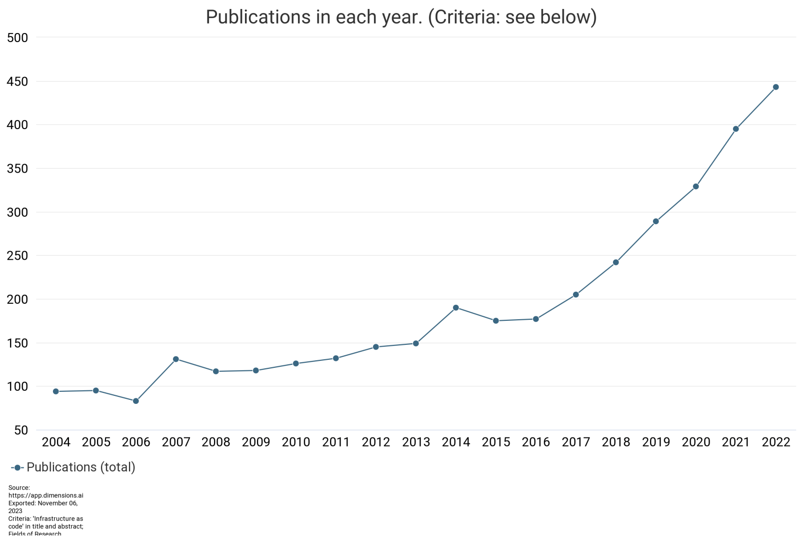

Figure 2.2: Dimensions Publications per year for Topic 'Infrastructure as Code'

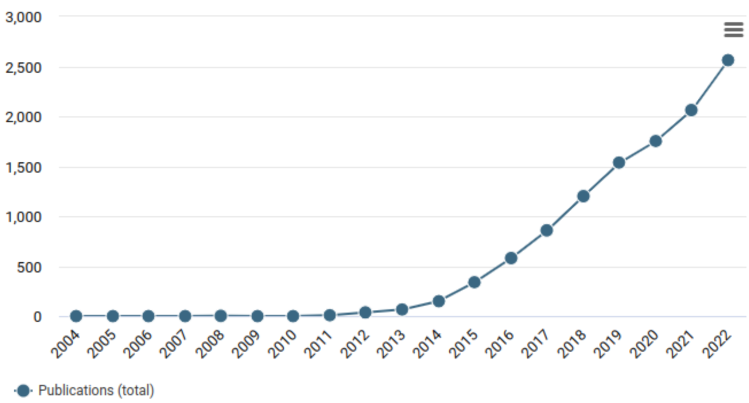

Figure 2.3: Dimensions Publication per year for Topic 'DevOps'

2.2.2 Part 2: A Case Study of IaC Implementation

The second part of the thesis uses a single-case study approach to explore IaC implementation in a real-world setting. Through the use of semi-structured interviews conducted with people interacting with IaC at various I aim to achieve the following objectives:

- Analyze how software practitioners within the case study organization evaluate the efficiency of their IaC implementation.
- To identify challenges and benefits associated with IaC implementation
- To compare the challenges and benefits associated with IaC implementation, as experience by the case study participants, to the broader literature on the topic.

2.3 Motivation behind research

Comparing figures 2.2 and 2.3 we can see that Infrastructure as code is a relatively new field of research with the majority of the research focusing on DevOps instead. However, with the increasing popularity of cloud-based architectures, there is a growing need for more research on this topic.

There are two main reasons for the motivation behind this thesis:

- Gap in research - First, there is a gap in white literature on IaC in regard to the best metrics for measuring efficiency. This makes it difficult for companies to get ideas on the quality of their infrastructure, and identify areas of improvement. Secondly, most of the research revolves around a specific Framework or Tool for either implementing IaC or extending a functionality of IaC. [38]
- Practitioner focus - Most of the available white literature does not focus on the practical aspects. Only 21% of studies found by Rahman et al. [38] revolved around empirical studies. As such, in this thesis I plan to conduct a Multi-Vocal Literature review that will serve as the first part of this two-semester long view into the topic of infrastructure as code. The conclusion of this thesis, along with the created artefact will serve as starting point of the second's semester paper - an empirical study, exploring a specific case inside a company called Jumia. This way, I aim to reduce the gap between grey literature and white literature and provide information usable by both researchers and practitioners.

2.4 Research question

Thus for each parts of the thesis I formulate the following research questions:

2.4.1 Part 1: MultiVocal Literature Review

- RQ1: How do software engineering practitioners define "efficiency" of Infrastructure as Code implementations?
- RQ2: What are the key metrics, reported in the literature, to evaluate the efficiency of IaC implementations?

2.4.2 Part 2: Case Study

- RQ3: How do software practitioners expect to evaluate the efficiency of IAC implementations?
- RQ4: How do the challenges and benefits associated with IaC implementation, as experienced by Jumia, compare to the broader literature on the topic?

Chapter 3

Methods - MultiVocal Literature Review

This chapter will provide an overview of the research methodology used in the first part of this project, a MultiVocal Literature Review

After conducting an initial search on available white literature on the IaC topic, I was unable to find a very significant amount available. Therefore, instead of conducting a Systematic Literature Review, a Multi-Vocal Literature Review (MLR) was considered instead.

The motivation behind choosing the MLR is discussed in more detail, accounting to the guidelines for conducting an MLR[58] in chapter 4.1.1

As defined by Ogawa et al. [37], "Multivocal literatures are comprised of all accessible writings on a common, often contemporary topic. The writings embody the views or voices of diverse sets of authors (academics, practitioners, journalists, policy centers, state offices of education, local school districts, independent research and development firms, and others). The writings appear in a variety of forms. They reflect different purposes, perspectives, and information bases. They address different aspects of the topic and incorporate different research or non-research logics".

MultiVocal Literature Review is a type of Systematic Literature Review but expanded to include both white literature and MultiVocal Literature (also called grey) literature. By doing so, MLRs incorporate multiple voices and perspectives, including those from academics, practitioners, journalists, and policy centres. MLRs are useful both to practitioners and to researchers, as they include both state-of-the-art research and state-of-practice in the given area.

There have been multiple previous successful applications of MLR in the Software Engineering area, [55] including in the DevOps area, such as [2]. There have

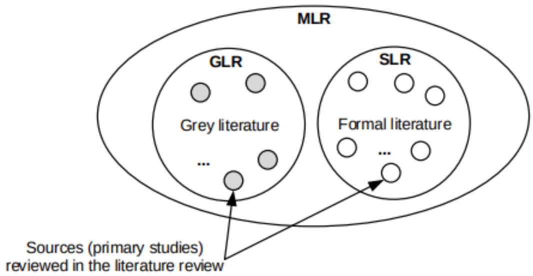

Figure 3.1: Venn diagram showing the relationship of SLR, GLR and MLR studies as per Garousi et al. [58]

also been some papers that combined MLRs with interview-based case studies[14]. To the best of my knowledge, this is the first MLR conducted on Infrastructure as Code specifically.
By conducting this MLR, I aim to complete three objectives:

- To do an overview of the current state of the literature regarding IaC
- To define efficiency in the context of IaC
- To gather IaC metrics related to efficiency
- To construct an artefact that will serve as the basis for achieving the main objective of this project: Finding which metrics DevOps practitioners prioritize when seeking to measure efficiency improvements gained through the adoption of IaC.

In order still maintain a high quality of research, I must choose the kind of grey literature used, as not all grey literature is created equal.

As Giustini et al. [15] noted, Literature is not White or Grey but a spectrum, from High Credibility/ High outlet control media (White Literature) such as books and government reports to Medium Credibility/Medium outlet control media (Grey Literature) such as Wiki articles, Q/A websites, newspapers to Low Credibility/Low outlet control media (Black Literature), such as opinions, tweets, Facebook posts.

For the purpose of this MLR, I will consider only white and some grey literature. Besides excluding all black literature, I will also exclude some types of grey literature, namely Q/A websites and Audio-Video media. The reason behind the exclusion of these types of Grey literature revolves around the depth of information. These types of media are often short, presented in an informal and condensed matter, and are often based on other types of literature such as blogs or white literature.

"White" Literature	"Grey" Literature	"Black" Literature
Published journal papers	Preprints	Ideas
Conference proceedings	e-Prints	Concepts
Books	Technical reports	Thoughts
	Lectures	
	Blogs	

Table 3.1: Spectrum of literature, adapted from [15]

While the benefits in knowledge by including grey literature are high, I must still be wary of the inclusion of such media to skew the results, as it is based on opinions and experiences that are not peer-reviewed. To mitigate this, I plan to follow a set of guidelines and procedures as described by Garousi et al. [58]

The procedure outlined by Garoussi et al. is an adaptation of the SLR guidlines presented by Kitchenham and Charters [21].

As such, MLR is done in three main phases, each with a number of subphases.

Planning the MLR (section 4.1)

- Motivating the need for an MLR (section 4.1.1) - based on a number of factors
- Defining the MLR's goal and raising the research questions (section 4.1.2) - matching the specific needs of the target audience while being as objective as possible

Conducting the MLR (section 4.2)

- Search process and Source selection (section 4.2.1) - through defined search strings and via both from full-text databases with more limited coverage (ex. IEEEXplore) and search engine (ex. Google.com)
- Quality assessment of sources (sections 4.2.2 4.2.3) - to ensure validity and objectivity
- Data extraction (sections 4.2.4 4.2.5) - as much as needed to address each RQ sufficiently
- Data synthesis (section 4.2.5) - through a systematic map

Reporting the MLR (section 4.3)

- Similar to the reporting phases of an SLR, described in the guidelines of Kitchenham et al. [21], but with the addition of two key points: different reporting style for different audiences and ensuring usefulness to the target audience.

Chapter 4

IaC MultiVocal Literature Review

As mentioned in the previous chapter, in order to maintain a high level or rigor, I adhere to the guidelines presented by Garousi et al. [58]. Each section in this chapter will discuss one of the three main phases, while each subchapter will delve into the specific subphases.

The complete process undertaken with this MLR is showcased in figure 4.1 below.

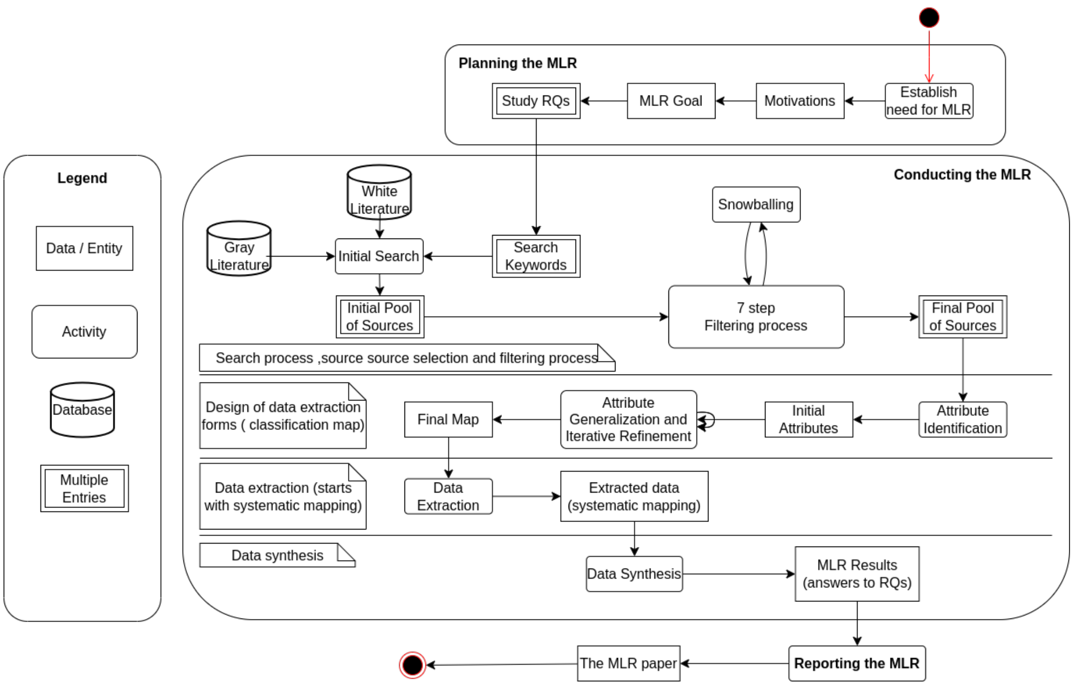

Figure 4.1: MLR process diagram

4.1 Planning the MLR

The first phase of conducting the MLR concerns its planning. First, a motivation behind the MLR is presented, followed by the research questions, and their specific objectives.

4.1.1 Motivation

Before conducting the MLR, it is important to ensure the need for it. According to the guidelines presented by Garousi [58], this is done systematically using a well-defined set of criteria/questions. In order to answer the questions, first a literature discovery should be undergone. This will serve as the basis for answering the questions.

Table 4.1 contains the list of questions/criteria and is used as a synthethised decision aid. One or more "yes" responses suggest the inclusion of Grey Literature, and, the higher the number, the higher the need for conducting an MLR in the topic. These criteria, as explained by Garousi et al., originate both from sources containing guidlines for conducting MLRs in other fields, and from the expereri-ence of the authors.

4.1.2 Research Questions

The goal of this MLR is to review and synthesize both the state-of-the art and practice in order to define efficiency in the Infrastructure as Code topic, and list primary metrics related to efficiency in the context of Infrastructure as Code.

Based on this, after a brief investigation into both White and Grey literature, the research questions presented in 2.4 were identified. These will also be used as the goal for this MLR.

4.2 Conducting the MLR

The second, and main phase of an MLR concerns its actual conduction. This is comprised out of a couple of steps: identification, filtering, quality assessment, data extraction and data synthetization of relevant literature.

4.2.1 Source Selection and Search Keywords

The search process, conducted in November 2023, was done using pre-defined search strings.

In order to identify the search strings, the guidelines presented by Kitchenhaim

#	Question	Answer
1	Is the subject "complex" and not solvable by considering only the formal literature?	Yes, most of the current formal literature revolves around DevOps as a whole. Besides, the formal literature cannot represent all the intricacies of real-life IaC implementations.
2	Is there a lack of volume or quality of evidence, or a lack of consensus of outcome measurement in the formal literature?	Yes. There is a lack of volume and quality of evidence relating to IaC metrics. Figure 2.2 showcases the limited amount of papers.
3	Is the contextual information important to the subject under study?	Yes, as IaC is also influenced by the organizational contexts that may not be fully captured in formal literature
4	Is it the goal to validate or corroborate scientific outcomes with practical experiences?	Yes, the goal of the paper is to define efficiency in the context of IaC, define list of metrics, while also validating and prioritizing them in a professional setting.
5	Is it the goal to challenge assumptions or falsify results from practice using academic research or vice versa?	No, while it is known that IaC has a positive outcome, there have not been many studies done on the actual metrics and their link to measuring this positive outcome.
6	Would a synthesis of insights and evidence from the industrial and academic community be useful to one or even both communities?	Yes, this knowledge will be useful both to practitioners to better adapt their solutions, and also to academics to further research in this area, as there is a marked lack of it.
7	Is there a large volume of practitioner sources indicating high practitioner interest in a topic?	Yes, there is a substantial volume of practitioner sources in Infrastructure as Code,along with high interest, as seen in 2.1

Table 4.1: List of questions whether to include Grey Literature, adapted from [58]

[21] were followed. This was an iterative process, each iteration being an exploratory search revealing more and more relevant search strings. One deviation from Garousi's guidelines is the separation of search keywords between literature types. This was done because grey literature search engines do not allow such concise filtering. On the other hand, a more detailed filtering was possible on academic databases.

- **Search String:** ("Infrastructure as Code") AND ((metrics OR efficiency OR measurement OR performance) OR ("adoption" OR "efficiency evaluation" OR "performance benefits")), written after 2015, in the Computer Science subject
- **Datasets:** Springer Link, ACM Digital Library,IEEE Xplore, Science Direct and Google.com

Table 4.2 contains the search strings for all of the sources, adapted to their respective query requirements.

Database	**Search string**
Springer Link	("Infrastructure as Code") AND ((metrics OR efficiency OR measurement OR performance) OR ("adoption" OR "efficiency evaluation" OR "performance benefits")) Additional Filters: Published between 2015-2023, written in English, Computer Science discipline
ACM Digital Library	[[All: "infrastructure as code"] OR [All: iac]] AND [[All: metrics] OR [All: efficiency] OR [All: measurement] OR [All: performance] OR [All: "adoption"] OR [All: "efficiency evaluation"] OR [All: "performance benefits"]] AND [E-Publication Date: (01/01/2015 TO 12/31/2023)]
IEEE Xplore	("Infrastructure as Code") AND ((metrics OR efficiency OR measurement OR performance) OR ("adoption" OR "efficiency evaluation" OR "performance benefits")) Additional Filters: Published between 2015-2023
Science Direct	("Infrastructure as Code") AND ((metrics OR efficiency OR measurement OR performance) OR ("adoption" OR "efficiency evaluation" OR "performance benefits")) Additional Filters: Published between 2015-2023, Computer Science subject area
Google.com	("Infrastructure as Code") AND ((metrics OR efficiency OR measurement OR performance) OR ("adoption" OR "efficiency evaluation" OR "performance benefits"))

Table 4.2: Search strings used for each database

4.2.2 Inclusion and Exclusion Criteria and their application

The source selection sub-phase deviates from Garousi's guidelines [58]. This was done to increase the rigour of the process, as there is a marked difference between the quality and content of grey and white literature. For example, grey literature does not usually have abstracts or is published in conference papers. The academic literature has five inclusion criteria and three exclusion criteria. Due to the formal requirements of this report, the papers must be written in English, and be accessible using AAU's licence. This also contains freely accessible literature. They must part of the computer science domain, and must discuss IaC. Otherwise, they must not be duplicates, or discuss specific tools or frameworks. The reason behind this is because this MLR is taking a tool-agnostic viewpoint. Besides, whole books have not been considered due to the time constraint of this project.

On the other hand, grey literature has the same inclusion criteria as Academic literature, but due to the lack of uniformity, and the possibility of false information, advertisements are not considered. Tutorials are also excluded as the MLR is not aiming at looking at specific tools. Besides, videos are not considered due to time constraints with this project. Table 4.3 contains the inclusion and exclusion criteria for white literature, while 4.4 contains the inclusion and exclusion criteria for grey literature.

Inclusion criteria	Exclusion criteria
Written in English	Duplicates
Published after 2015	Tools & Frameworks
In the computer science domain	Whole Book
Accessible using AAU's license	
Must discuss IaC	

Table 4.3: Inclusion and exclusion criteria for White Literature

Inclusion criteria	Exclusion criteria
Written in English	Duplicates
Accessible with AAU's license	Tutorials
Published after 2015	Videos
Must discuss IaC	Advertisements
In the computer science domain	Q/A websites
	Audio-Video media

Table 4.4: Inclusion and exclusion criteria for Grey Literature

4.2.3 Filtering of Relevant Sources

This section showcases all the steps and filters that were used in order to identify relevant sources. By developing on the process described by [21], a seven step process was used to discover and filter the literature.:

1. **Filter 1:** Dataset filtering based on strings
2. **Filter 2:** Dataset filtering based on inclusion and exclusion criteria
3. **Filter 3:** Dataset evaluation based on abstract and title
4. **Snowballing:** Back-and-forward reference search
5. **Filter 5:** Duplicates removal
6. **Filter 6:** Quality assessment
7. **Filter 7:** Full-text document assessment

This process is also showcased in figure 4.2 below.

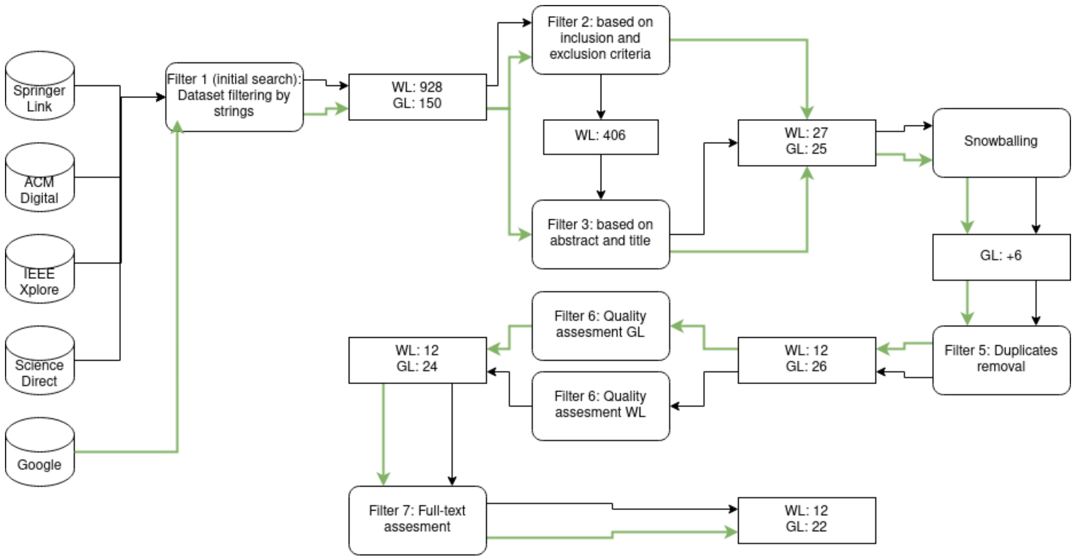

Figure 4.2: Filtering process diagram

Filtering of Academic Literature

In the case of the academic literature filtering, starting with the initial Filter 1, a search was conducted using the search strings defined in table 4.2. This yielded 928 search results.

The second step in the process was the application of Filter 2 over the 928 search results. This further limited our search results to 406 results.

After eliminating non-relevant literature, Filter 3, a dataset evaluation based on abstract and title followed. In the case of ambiguous abstracts, the summary and conclusion chapters were taken into consideration. This yielded 27 search results.

After eliminating unrelated research papers, all the remaining literature content was skimmed over, leading to the snowballing step. A back-and-forward reference search is conducted leading to no addition of papers.

After discovering all relevant literature, it was passed through Filter 5 in order to remove all duplicates, leading to 12 remaining publications.

Next, To ensure a minimum level of quality, Filter 6 concerns a short quality assessment, as described by Kitchenhaim et al. [21], of the academic literature to ensure its credibility and accuracy.

As such, In order to pass filter 6, all papers must have Limitations and Validity concerns discussed, must be formally published in reputable sources and the results and methodology clearly explained. Compared to Grey literature, academic papers are already considered to be of a higher quality, due to being peer-reviewed and published, and as such, the quality assessment was not done as extensively.

In the end, 12 white literature publications remain for full-text document assessment.

Filtering of Grey Literature

In the case of Grey literature, a more thorough quality assessment process is required, as Grey literature does not have the same rigor as white literature. This subchapter presents the filtering process done in the case of grey literature.

Google.com was selected as the search database for this MLR. Starting with filter 1, a search is conducted using the search strings defined in table 4.2.As opposed to academic literature, where all publications were taken into consideration, in the case of grey literature, two considerations are done regarding when to stop the search, either when theoretical saturation is achieved, or when an efford bound of 150 search results of achieved. According to [58], these two considerations, besides evidence exhaustion are enough to conduct an MLR.

The second step in this process, Filter 2 concerns filtering based on inclusion and exclusion criteria. Due to the nature of grey literature, this filtering cannot be done automatically. As such, skimming the article was considered as necessary. As such, Filter 2 and 3 will be done concurrently.

Step 4, snowballing has lead to the addition of 6 pieces of literature. The next step in this process was the removal of duplicates, yielding to a total number of 26 publications.

To ensure a minimum level of quality, and to maintain the quality of this MLR, the quality assessment criteria of grey literature for software engineering defined by Host and Runeson, and adapted by [58] were used as the base for this quality check. The Checklist presented by Garoussi is very in-depth, the author remarking that he has not seen any MLRs using it. Nonetheless, I use a more bare-bones version of it in this MLR. Table 4.5 showcases the 5 categories and the 8 questions that comprise it. One category that was omitted is Methodology. The reason behind this is the fact that none of the grey literature included in this MLR discusses it. Thus, no differentiation can be done based on it.

In order to verify the author's expertise and credentials and the reputation of the publishing organization I have used LinkedIn. To validate objectivity I looked out for any self-promotion or vested interest. The novelty of the publication, I verify if it establishes or creates something new, or poses some unique views. Finally, grey literature is graded based on the outlet type tier. One difference to the criteria presented by Garoussi [58] is the consideration of Blog posts as 2nd tier, instead of 3rd tier, thus awarding it a measure of 0.5 instead of 0.

To assess the final quality of the papers, for every question a positive response is awarded 1 point, while a negative response is awarded 0 points, with the exception of the Outlet type category, where half points are also awarded, and the question regarding vested interest, where a negative response receives is awarded 1 point. All points are aggregated at the end, and out of a total of 16 points (100%), a validity threshold of 8 points (50%) is required for a grey literature paper to pass this filter. This filter was passed by all but 2 publications.

Criteria	Questions
Authority of the producer	• Is the publishing organization reputable? (eg: Google, Software Engineering Institute) • Does the author have expertise in this area?
Objectivity	• Is the statement in the sources as objective as possible? • Is there vested interest?
Date	• Does the item have a clearly stated date?
Novelty	• Does it enrich or add something to the research? • Does is strengthen or refute a current position?
Outlet Type	• 1st tier GL (measure=1): High outlet control/ High credibility: Books, magazines, theses, government reports • 2nd tier GL (measure=0.5): Moderate outlet control/ Moderate credibility: Annual reports, news articles, presentations, videos, Q/A sites (such as StackOverflow), Wiki articles • 3rd tier GL (measure=0): Low outlet control/ Low credibility: Blogs, emails, tweets

Table 4.5: Quality Assesment Criteria for Grey Literature

4.2.4 Final Pool of Sources

In total, I have identified 16 relevant artefacts from Grey literature, with an addition of 6 artefacts from snowballing, totalling 22 Grey literature artefacts. Additionally, I have identified a total of 12 artefacts of White literature. Table 4.6shows a complete overview of the number of papers remaining after each step in the filtering process. Figure 4.4 shows the cumulative number of filtered sources per year. As seen, the last two years, 2022 and 2023 show a significant increase in both Grey and White literature published. A similar trend can also be seen for DevOps, which Infrastructure as Code is part of. [59] However, IaC, compared to DevOps still remains a relatively understudied topic. [38]

A distribution of the final pool of sources per database can be seen in figure 4.5. This dataset is comprised of 64.7% Grey literature and 35.3% White literature. Grey Literature was sourced from Google.com, while White literature was obtained from ACM Digital (8.8%), IEEE Xplore(8.8%) and Science Direct (17.6%)

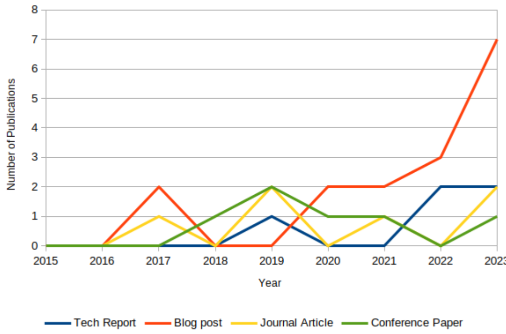

Figure 4.3: Final pool of sources per source type

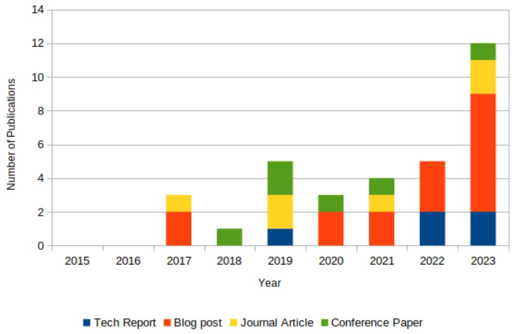

Figure 4.4: Final pool of sources per source type

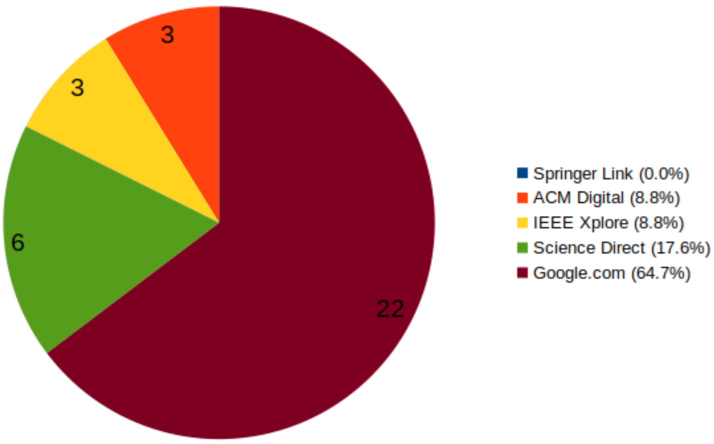

Figure 4.5: Distribution of final pool of sources per database

Database	Filter 1	Filter 2	Filter 3	Snowball	Filter 5	Filter 6	Filter 7
Springer Link	551	59	5		0	0	0
ACM Digital	234	215	5		3	3	3
IEEE Xplore	51	47	6	+6	3	3	3
Science Direct	92	85	11		6	6	6
Google.com	150	94	25		26	24	22
Total	1078	500	52	58	38	36	**34**

Table 4.6: Filtering Process in numbers

4.2.5 Data Extraction and Qualitative Synthesis

To efficiently extract relevant data from the dataset, a Systematic map was created. Each research question was analyzed, and attributes and aspects that would answer it were identified. Afterwards, a small subset of the dataset was tested against this systematic map, and through an iterative process and attribute generalization the final extraction form was created, following the guidelines presented by Garoussi et al. [58]. The systematic map is shown in figure 4.7.

RQS	Attribute/Aspect	Categories	(M)ultiple / (S)ingle
-	Source Name	Name	S
-	Source Type	Grey Literature, White Literature	S
1	Definition	Efficiency	S
1	Quote	Quotes when discussing efficiency	M
2	Metrics	Metric Name	M
2	Metrics	Purpose of metric	M

Table 4.7: Systematic map

To make the data extraction process more rigorous and increase traceability, I also logged attributes such as Date, Database, Article name, Hyperlink and Data accessed for each paper.

4.3 Reporting the MLR

This section follows the last phase of the MLR, and concerns assessing RQS1 and RQS2 based on the results of the data extraction in detail.

4.3.1 RQ1:How to define "efficiency" of Infrastructure as Code (IaC) implementations?

SUSE, an open-sources software company defines IT efficiency as "the primary goal of most IT organizations, and is fundamentally considered to be 'doing more with less' by maintaining or improving the quality of IT services that is delivered to the organization, while reducing costs" [53]. In alignment with this definition, this MLR has looked to define efficiency in the context of Infrastructure as Code. To do so, we have looked at all the benefits and recurring patterns behind IaC adoption.

However, while speed and cost are primary indicators, it's crucial to recognize that the outcome of efficiency in IaC is not solely confined to these .The holistic view also considers the following outcomes: "Scalability and Standardization," "Security and Documentation," and "Disaster Management."

Speed Increase and Cost Reduction

Highlighted in 84% of the examined papers(11 out of 13), Speed Increase and Cost Reduction are central aspects of IaC efficiency outcomes.

One of the ways that IaC can achieve "decreased time to market", "resilience" and "speed" [27] [43] is by "eliminating manual processes" [43]. By using tools such as Terraform, infrastructure can be provisioned automatically, across multiple environments. Moreover, by treating infrastructure as code, there is a reduction in the risk and effort in making changes to the infrastructure. [24] [34]. The ability to introduce versioning to the infrastructure also allows multiple people to work on it more easily and promotes a more collaborative and efficient team relationship between DevOps and SecOps. [18]. Additionally, the ability to template infrastructure eliminates the need to do boilerplate "routine, repetitive things". [35] through the use of variables.

This ability to more easily orchestrate and provision infrastructure also has a positive impact on cost, as noted by [6] and [18]. The "as-code" traceability benefit also lowers the technical debt of change, lowers the possibility of human error, and overall leads to lower operation cost and time spent on manual repetitive tasks. This speed increase and cost saving is also mentioned in papers [17], [45], [32], and [51].

In summary, IaC through the use of variables, templates and versioning control reduces the time it takes to set up, configure and manage infrastructure. Also, by eliminating repetitive and time-consuming tasks, IaC reduces the need for manual labour. Besides, these features also lead to a less error-prone infrastructure,and implicitly to fewer issues that can incur additional costs.

Scalability and Standardisation

Another outcome of efficiency of IaC involves the ability to scale infrastructure resources dynamically and maintain standardized configurations across diverse environments. This outcome ensures adaptability to varying workloads and guarantees consistency in deployments, and was mentioned in 69% of papers (9 out of 13).

One of the benefits of IaC, as mentioned by Kief Morris, author of book titled Infrastructure as Code is "Promotion [...], moving things from one environment to the next"[35]. Through the use of variables and reusable templates, environments can be easily changed, created, and promoted, bringing standardisation between multiple environments, as they are provisioned from the same code source. This "system consistency"[24] / standardisation also "eliminates configuration drifts" [6] [18], ensuring applications behave in a predictable manner over multiple environments.

Through IaC, one can also automate the configuration of resources for defined infrastructure, allowing dynamic scaling based on demand or other factors. This automatic elasticity is especially beneficial for applications with high demand fluctuations, such as e-shops. This benefit of "increased scalability and flexibility." was also mentioned in papers [32] and [51]

To summarize, IaC improves scalability through its dynamic resource provisioning and improves standardisation through parametrizable reusable templates that promote both consistency and enforcement of policies.

Security and Documentation

As mentioned in 53% of papers (7 out of 13) another way to measure efficiency improvements brought by the adoption of IaC is through the more robust security measures and comprehensive documentation.

Security is promoted by IaC through its added transparency and traceability, more easily enabling developers to identify security flaws and fix them. Besides, the benefit of "-as code" also enables the treatment of infrastructure as normal code, enabling ease of documentation.

IaC implements the concept of immutable infrastructure, where server instances are replaced instead of updated. This streamlines the infrastructure, with vulnerable components being replaced with newer secure ones, reducing the attack surface.

IaC "improves security strategies" [32] [51], as it supports integration with automatic security testing tools to do vulnerability, unit and integration tests. This " helps people who are concerned about [the risks of automation] and worry that things can be done wrong automatically." [35]. Paper [27] also mentions this benefit of IaC "enabling accelerated deployment through repeatable automated testing."

IaC also allows for traceable and auditable changes to code. All modifications are recorded in version control, providing transparency and rollback capability, while also facilitating easier knowledge sharing. Besides, through the use of "repeatable templates aligned to organizational security standards." [6] [18] security configurations can be implemented consistently across the entire infrastructure.

The IaC code also serves as a form of self-documentation, as team members can easily learn the architecture, dependencies and configuration from reading the code.

In conclusion, IaC improves security by promoting the immutable infrastructure concept, enforcing consistent configurations deployed through templates, and facilitating automatic compliance checks and easy rollback. At the same time, IaC also improves documentation by serving as version-controlled anotable self-documenting code.

Disaster Management

The outcome of improving the efficiency of IaC can also be measured through better disaster management practices, as mentioned in 6 papers(46% of papers).

IaC "helps with recovery and setting up new testing environments" [43], as infrastructure can quickly be rebuilt according to predefined configurations, ultimately resulting in a better recovery time objective. Moreover, as mentioned in [24] and [34] IaC also brings "Reduced risks and downtime" and "Faster troubleshooting," through version control. In the event of a disaster, a versioned backup can be deployed, ensuring a rollback to a known stable state, while the failed version can more easily be debugged.

Moving beyond immediate recovery, into disaster prevention, IaC also allows infrastructure configuration validation, in order to catch misconfigurations that might lead to disaster scenarios. As mentioned in [17]", by eliminating manual configuration errors and ensuring consistency, organizations experience greater reliability, stability, and reduced downtime." The concept of immutable infrastructure,that IaC promotes, also helps in disaster recovery, as failed components can just be replaced, instead of debugged. As such, through the use of IaC "defects and updates are addressed in a more efficient manner through shorter cycles, tight collaboration and frequent Updates" [27]

In conclusion IaC has a very important role in disaster recovery by reducing downtime (through easier and quicker redeployment of artefacts), enabling quick deployment of versioned backups, while also helping with disaster prevention through configuration validation and versioned source code, and by promoting the immutable infrastructure concept.

RQ1: Definition of Efficiency in the context of IaC: In the context of IaC, efficiency refers to the ability to streamline and manage the infrastructure through automated processes, ensuring consistent and repeatable deployments across various environments, reducing errors and risks, and improving disaster management, while saving time and costs. Efficiency in IaC is also marked by the ability to enforce security and improve documentation through self-documenting code-based modular configurations. This decreases the need for manual configuration, limiting configuration drifts, and shortening the time needed for scaling while also leading to faster recovery.

Alternatively, one could consider a more conventional definition of efficiency, by looking at efficiency in software engineering.

> "Efficiency in software development refers to the ability of a system to maximize performance and productivity while minimizing resource usage. "[39]

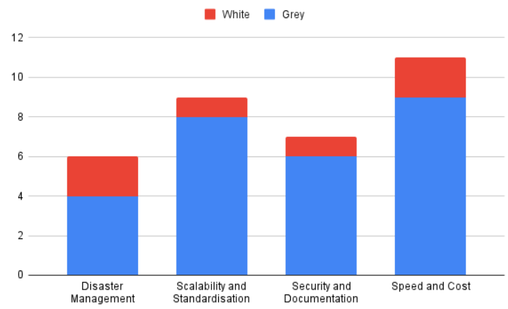

Figure 4.6: Final pool of sources per source type

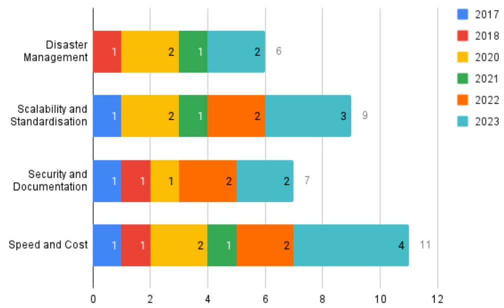

Figure 4.7: Final pool of sources per source type

RQ1 Data analysis: Out of a total of 34 publications, 13(38.23% of papers) of them discussed efficiency or benefits relating to IaC. Figure 4.6 shows the ratio of Grey to White literature, and the amount of papers that discussed each of the four outcomes presented.

Figure 4.7 shows the distribution of papers per year that discussed each of the four outcomes. The year with the most publications that discussed any of the outcomes is 2023, this being in line with Figure 2.2 and 2.1 showcasing the increase in attention that IaC receives, and figure 4.4 showing the number of publications per year used.

4.3.2 RQ2:What are the key metrics used to evaluate the efficiency of IaC implementations?

This subchapter presents findings addressing the second research question: "What are the key metrics, reported in the literature, to evaluate the efficiency of IaC implementations? " Examining the sources, from a total of 34 papers, 14 of them (41.17% of total papers) discuss metrics.

Metrics

Below I present the metrics that were identified in this MLR, along with the times they were mentioned and a description.

Lead Time for Change

Mentioned in 9 studies[16], and one of the four main DORA metrics[49][50] [17]Lead Time for Change records the time it takes to implement, test, and deploy the code [54], and used as velocity metric [9] of how well an organization meets its goals.[34]. A case study also remarks that Lead Time was one of the metrics with a "need to track" [46] A well-collaborating team can be linked to a low Lead Time for Change. [3]

Time to Restore Service

Mentioned in 9 studies, and one of the four main DORA metrics[49][50] [17] Time to Restore Service (TTRS) metric measures the average time taken to restore a system after a failure.[4] This metric can also be posed as a question: "Once you make a mistake, how long does it take to fix the problem?"[54] TTRS can be also used as a velocity metric [9] of how well an organization meets its goals.[34], It was also remarked in a case study that Time to Restore Service is a metric that needs to be tracked. [46]. Besides TTRS can be linked directly to greater customer satisfaction, faster application delivery, and better cost control. [3]

Change Failure Rate

Mentioned in 8 studies, third of the four main DORA metrics[49][50] [17], used to measure velocity [9]of how well an organization meets its goals[34], Change Failure Rate allows us to analyze whether the automation and optimization operations implemented are really effective: that is, they do not produce an excessive number of errors,[54] or fail to to respond to requests appropriately [13]. Monitoring tools such as Kinesis and Graylog are useful when measuring this metric. [31]

Deployment Frequency

Mentioned in 8 studies,often discussed by practitioners [46], and the fourth main DORA metrics[49][50] [17], the Deployment Frequency metric counts the number of releases in a given time period [54], and is used as velocity metric [9] of to how well an organization meets its goals.[34] One thing to note is operations teams often prioritize server uptime whereas developers favour release frequency.[29]

Deployment Time Duration

Mentioned in 5 studies, in different forms and granularity, Deployment Time Duration [54] is considered to be the time it takes for the infrastructure to get provisioned with the new deployment settings. This can either be measured with high granularity through the Plan, and Apply Run Times (in Terraform) [9][3], or overall, through the total Provisioning Time [4].

Uptime and Availability

Mentioned in 3 studies, the Uptime and Availability metric, either overall availability of servers,[31], or per server uptime [29], and should be more than 99.999% of the time for users of the system, with a ratio based on availability, reliability, and uptime.[3]

Idle Percentage (also knows as saturation)

Mentioned in three study in different forms, [13] the Saturation metric concerns the "saturation" of the infrastructure, or how much of the infrastructure that is provisioned is currently not used. While having a small percentage of infrastructure idle, to serve as scaling buffer (as automatically scaling infrastructure takes time to spin up additional servers) is a good idea, there is a downside in cost when keeping too much infrastructure idle. This can be measured either by checking the memory and CPU usage [9], or Disk usage [31], on a per-server basis, or by aggregating these metrics and looking at the overall infrastructure.

Number of Drifts

Mentioned in 2 studies, [4] [9] the Number of Drifts metric provides a sense of "how stable the infrastructure is without having unexpected post-deployment modifications."

Ratio of Drifted/Total Resources

Mentioned in 2 studies [4] [9] the ratio of Drifted to Total Resources measures the percentage of infrastructure components drifting from their intended configuration over time. This can lead to an increase in security vulnerabilities, performance issues, and general system instability.

Wait Time

Mentioned in one study [3], Wait Time is the duration that a work item spends in a non-productive environment. Wait time falls in the outcome "Speed Increase and Cost Reduction", underscoring the impact on operational efficiency and the need to minimize delays during task executions.

Team Satisfaction

Mentioned in one study [3], and considered to have low relevance, Team Satisfaction / Happiness is a subjective metric pointing toward overall team satisfaction. While not directly linked to IaC, it can be argued that a well-working infrastructure has a positive impact on the team using it.

WIP Load

Mentioned in one study [3], the Work In Progress Load metric measures the number of artefacts that are currently being worked on. Not directly linked to IaC, but in a similar vein to WIP Limit in Kanban, the WIP Load metric is used as measure against overloading the team working with IaC.

Unplanned Work rate

Mentioned in one study [3], Unplanned Work Rate measures the amount of time spent on unplanned work. A high Unplanned Work Rate points to an IaC deployment that lacks quality, especially if the reason behind the UWP is an unexpected outage, while a low UWR indicates better predictability and reliability in the deployment process. [3]

Deployment size

Mentioned in one study [3], the Deployment Size metric tracks the size of deployment artefacts shipped to production. Large deployments done once in a while is seen as counterproductive, as it can more easily introduce bugs, compared to deploying small and often. DORA report 2021[49] shows and recommends that elite DevOps teams deploy small and fast, multiple times per day.

Test case errors per push

Mentioned in one study [9] a high number of Test Case Errors per Push can be linked to a poor quality of IaC code or an inefficient test execution, with tests timing out due to lack of resources or lack of optimization.

Cycle Time Value

Mentioned in one study, [3] Cycle Time Value measures the time it takes for an artefact to go from idea, through acceptance,build,test and stage to production. A high Cycle Time Value can be linked to increased friction between developers and the business. [3]

Infrastructure Code Coverage
Mentioned in one study, [4], the Infrastructure Code Coverage metric is used to measure the percentage of infrastructure components managed by IaC compared to the total number. High infrastructure code coverage implies improved manageability, reproducibility, and automation capabilities for systems. [4]

Documentation update frequency
Mentioned in one study, [4], the Documentation Update Frequency cab be measured by calculating the average frequency that documentation is updated relative to code or system changes. Stale or out-of-date documentation can lead to operational inefficiencies, onboarding issues, and system misuse. [4]

Drift detection scan time
Mentioned in one study, [9], Drift Detection Scan Time is a metric that provides indication of how well the test framework is scaling compared to the infrastructure. Drift Detection scans must be run often to limit infrastructure misconfigurations.

Mean Time to detection
Mentioned in one study, [3]. A high Mean Time to Detection, or the time it takes from the incident occurring to the team being informed and getting it fixed, points to a lack of efficiency of the monitoring system. Moreover, a high MTTD can create bottlenecks. [3]

Latency
Mentioned in one study, [13], Latency concerns the median time it takes to complete an action when using the infrastructure. This can help with finding bottlenecks, understanding which resources require the most time to access, and noticing when actions suddenly take longer than expected. [13]

Traffic, requests per second
Mentioned in one study, [13] the Traffic metric captures load or on the infrastructure. A constant high load can point to either a need to scale up, or that there is a problem which is preventing traffic from being routed correctly.

In conclusion, this MLR has revealed a total of 22 metrics related to IaC. From these five metrics stand out as the most frequently mentioned, indicating their prominent role in the evaluation of IaC efficiency. The top five metrics, based on their recurrence in the literature, are:

- Lead Time for Change (Mentioned 9 times)
- Time to Restore Service (Mentioned 9 times)
- Change Failure Rate (Mentioned 8 times)

	Speed Cost	Scalability Standardisation	Security Documentation	Disaster Management
Lead Time for Changes	X			
Time to Restore Service				X
Change Failure Rate	X		X	
Deployment Frequency	X	X		
Deployment Time Duration	X			
Uptime and Availability		X		X
Idle Percentage		X		
Number of Drifts		X	X	
Ratio of Drifted/Total Resources		X	X	
Wait Time	X			
Team Happiness				
WIP Load	X			
Unplanned Work Rate	X			X
Deployment size		X		
Test case errors per push	X		X	
Cycle Time Value	X			
Infrastructure Code Coverage		X	X	
Documentation update frequency			X	
Drift Detection Scan Time			X	
Mean Time to Detection			X	X
Latency		X		
Traffic, Requests/sec		X		

Table 4.8: Correlation between outcomes of efficiency and metrics

- Deployment Frequency (Mentioned 8 times)
- Deployment Time Duration (Mentioned 5 times)

A correlation between RQ1 and RQ2

To correlate RQ1 and RQ2, this table showcases all of the identified metrics, the four outcomes of efficiency in the context of IaC, and tries to find a correlation between the two based on the context in which they were discussed in the papers, and the experience of the author. While there might be some errors with some metrics belonging to different categories, next semester's case study will help rectify this. This table 4.7 will serve as an artefact in the next semester's case study on Jumia, with the role of guide during the interview phase.

For example, reading this table, the metric "Deployment Frequency" is useful

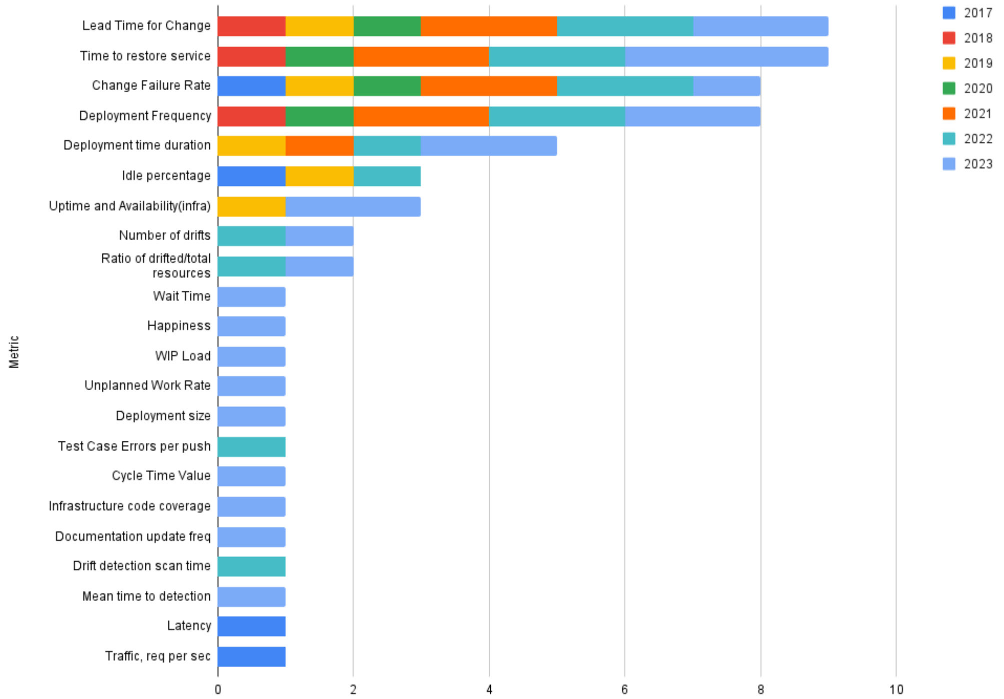

Figure 4.8: Metric mention per year

when the outcome is an improvement in "Speed and Cost" and "Security and Documentation".

RQ2 Data analysis: Figure 4.8 shows times each metric was mentioned per publication in that specific year. On the other hand, The year with the most publications that discussed any of the metrics is 2023, this being in line with Figure 2.2 and 2.1 showcasing the increase in attention that IaC receives, and figure 4.4 showing the number of publications per year used.

Chapter 5

Methods - Case Study

For the second part of this thesis, I will conduct an exploratory case study. The purpose of a case study is to understand and "investigate contemporary phenomena in their specific context".[41] The purpose of this case study is exploratory, aiming to discover what is happening and to seek new insights. [41]
The case study was conducted at Jumia, a leading e-commerce platform in Africa. As presented in 2.4, through this study, I aim to address the following two research questions:

- How do software practitioners expect to evaluate the efficiency of IAC implementations?
- How do the challenges and benefits associated with IaC implementation, as experienced by Jumia, compare to the broader literature on the topic?

Furthermore, to comprehensively answer the second research question, I will use the existing literature dataset from the MLR, along with an additional literature search.

5.1 Case selection

According to Yin, case studies can be chosen based on convenience, purpose and probability.[8] Moreover, single-case studies can be justified for reasons such as studying a critical case, a typical case, a revelatory case, or a longitudinal case. [8] In this study, the case falls under the category of "typical", chosen for it's convenience. [8] Baskarada explains that a convenience selection basis is expedient for data collection purposes.
The case study can be classified as "typical"for several reasons:

- Jumia's IaC implementation isn't neither at it's start or completion (0% or 100% IaC adoption), nor do they use tools outside of the industry standard.

This leads to the case not to be considered "extreme" case.[61]

- Jumia's approach is not "critical" or "revelatory". [61] They haven't revolutioned or pioneers a new IaC implementation strategy, but instead try to adhere to industry standards and guidlines. This leads to the case not being considered "paradigmatic", or prototypical.
- Since I am not studying the same case study at two different points in time, it also cannot be considered a "longitudinal case".[61]
- As such, Jumia's case is not unique, and by studying their experience we gain insights into common challenges and considerations faced by organizations who use the same guidelines and toolset. This alligns with the concept of a "typical" case where our "objective is to capture circumstances and conditions of an everyday or commonplace situation" [61]

The choice of Jumia as the subject for this case study comes from my position within the organization. As an intern in the DevOps department working as a Site Reliability Engineer, I have direct access to relevant information, people and insights into Jumia's implementation of IaC. For the sake of this case study, through my engagement with Jumia, I will gather qualitative data, in the form of interviews.

5.2 Case description

Founded in 2012, Jumia has rapidly grown to become a leading e-commerce platform in Africa, present in over 10 countries as of 2024. Often nicknamed "the Amazon of Africa", Jumia identifies three main pillars that support its growth.

- Jumia Marketplace: connects millions of consumers to thousands of sellers.
- Jumia Logistics: supports the delivery of millions of orders.
- Jumia Pay: a proprietary digital payment solution [19]

Consumers can buy products, order meals, purchase groceries, pay bills and make donations through Jumia's platform. But, despite it's growth, Jumia has yet to generate profit, with management focusing scaling the business through usage growth, and cost reduction. Part of this operational strategy is the adoption of IaC to automate the provisioning, configuration, and management of resources. One important thing to note is Jumia's achievement to reduce it's tech and software costs by over 28 % through optimizing the infrastructure and resource allocation. [19]
The implementation of IaC at Jumia uses several tools and methodologies:

- Packer: used to create AWS Amazon Machine Images (AMIs), that serve as a base OS for the EC2 instances and EKS nodes.
- Terraform: used to orchestrate the AWS provisioning and deployment, while also ensuring the elimination of manual workload while promoting consistency and documentation.
- Amazon EKS: also called Elastic Kubernetes Service, is used to host Jumia's applications across many verticals and environments. To note, Jumia usually has Testing, Staging and Production environments.

Aditionally, Jumia has also integrated GitOps practices in some of its workflows. This is done through:

- Atlantis: is an application for automating Terraform via pull requests, and is used as a way to migrate from self-managed Kubernetes clusters to EKS clusters, while also ensuring a lower change of drifts.

In order to ensure monitoring and continuous integration, Jumia also leverages:

- Grafana: used for monitoring of infrastructure metrics, and to address potential issues with the infrastructure
- Jenkins: used for automated continuous integration and delivery (CI/CD) pipelines

Currently, the SRE team in Jumia has approximately 30 people, spread into teams focused on different functions, or different verticals.
This case study will delve into the experience of Jumia's DevOps team in implementing IaC, the observations taken during my time as an intern, with the purpose ofanswering the two research questions presenter in chapter 5

To note, the 21 metrics defined in chapter 4.3.2 will serve as artefacts throughout this case study.

5.3 Data collection

Data collection techniques can be defined into three levels according to Lethbridge et al[26].

- First-degree methods - when where the researcher is in direct context with the subjects, and data is collected in real-time. Interviews and observations with "think-aloud protocols" fall in this category.
- Second-degree methods - when the researcher collects data without direct contact with the subjects. For example, monitoring the usage of software engineering tools falls in the indirect category.

- Third-degree methods - when the researcher does an independent analysis of work artefacts that are already available. Analyzing failure reports, or requirement specifications falls under this category.

Data collection: First degree methods, semi-structured interviews (open questions) There are multiple ways of collecting evidence in doing Case Studies. As identified by Yin et al. [60] most of them are qualitative in type. Qualitative data is non-numerical,usually in the form of opinions, values, beliefs, and behavior. So e of the ways include Direct Observations (eg. human actions or physical environment), Interviews (eg. open-ended conversations, semi-structured discussions), Archival records (eg. student records), Documents (eg. e-mails, letters, reports).[60]

Additionally, Runeson et al. [42] mentions interviews as being one of the most important and frequently used data collection ways for case studies."The reason for this is simply that much of the knowledge that is of interest is not available anywhere else than in the minds of the people working in the case being investigated."

For this instance, Interviews in the form of Semi-Structured Interviews were chosen due to several factors:

Firstly, semi-structured interviews allow more fluidity in the discussion, prompting the participant to elaborate in more details based on the topics being discussed, and as the conversation progresses.
Secondly, practitioners' experiences with IaC are often context-specific, influenced by specific technical environments. Semi-structured interviews can capture effectively these nuances which might be overlooked in more rigid/structured data collection.

5.3.1 Semi-structured Interviews

Runeson et al. [42] divides interviews into three main types, based on their objectives, unstructured, semi-structured and fully structured. Unstructured interviews are most feasible for exploratory studies, where only little is preconceived about the outcome of interviews. Fully structured interviews contain closed questions, asked in the listed order, with an explanatory objective.[42]. In semi-structured interviews, questions are planned and designed to be more open-ended, and not necessarily asked in the listed order. They are very common in software engineering case studies, such as this one. [42].

Besides their objective, interviews can also be divided based on three general principles. As shown in figure 5.1, the funnel model starts with open-ended questions, moving toward more specific ones. The pyramid model starts with specific questions, ending with open ones. The timeglass model is a mixture of the two, starting with open-ended questions, straightening the structure in the middle, and opens up again toward the end of the interview. [42].

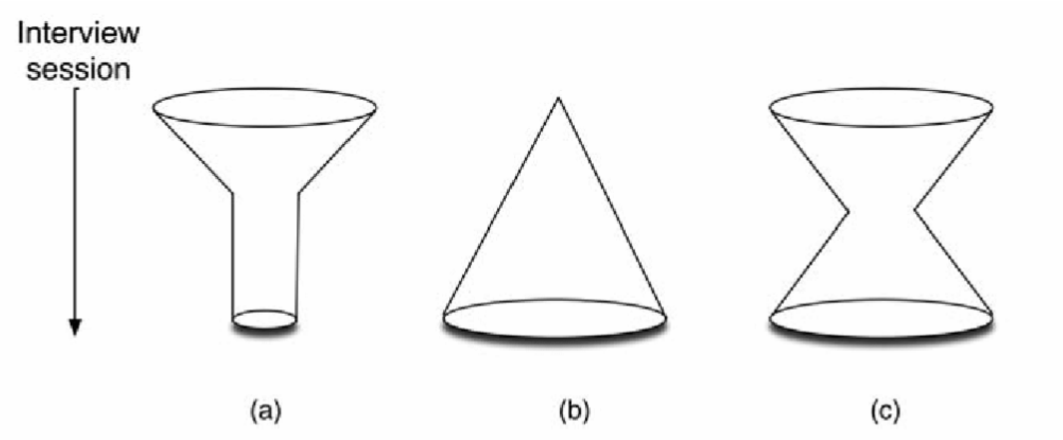

Figure 5.1: General principles for interview sessions: (a) funnel, (b) pyramid, and (c) timeglass as per Runeson et al. [42]

For this case study, semi-structured interviews using a timeglass model were selected. The reason behind this choice is the open-ended nature of semi-structured interviews offers flexibility, which bodes well with the exploratory nature of the case study. Additionally, the timeglass model was chosen because it allows for a mix of structured and unstructured questions: the questions regarding metrics start as semi-structured, but turn to fully structured(along with the presentation of a curated list of metrics to discuss) should the interviewee require guidance.

The semi-structured interviews were conducted with key members of Jumia's tech team, who both maintain, manage and fireman the infrastructure. The interview was structured into five categories: introductory, general IaC adoption and insights, prioritization of IaC metrics, metric measurement and tracking and impact on organizational efficiency.
The first two categories represent the broader sections of the hourglass model, the next two represent the narrower sections, and the final category broadens again.
The purpose of each part is as follows:

- The aim of the introductory questions(Part 0) were to obtain basic information and start the flow of conversation.
- Part 1 discusses the motivation behind adopting IaC, the observed benefits, efficiency in IaC and its impact on DevOps processes.
- Part 2 explores Jumia's currently established metrics, then presents a list of metrics to be discussed and prioritized, and asks about the factors influencing prioritization.

- Part 3 delves into how data for these metrics is collected, along with the tools and platforms that are used for monitoring and analysis, and the impact of measuring these metrics.
- Lastly, Part 4 looks into how IaC has benefited areas outside DevOps, and Jumia's ability to deliver value to customers, with specific examples.

The complete interview guide can be seen in section 5.3.1

Interview Guide

PART 0 – Introduction of Participant

Please introduce yourself briefly from a professional perspective. This includes your educational background, years of experience, and your role at Jumia.

Part 1: General IaC Adoption and Insights

1. Can you share with me the motivation behind adopting Infrastructure as Code (IaC) at Jumia? What was the background or driving force for this adoption?
2. What are the primary benefits you have observed from adopting IaC?
   - Follow-up: Can you provide specific examples of these benefits in practice?
3. How do you see efficiency in Infrastructure as Code (IaC), and how would you define it?
4. In your perspective, how has IaC impacted the efficiency of the DevOps processes?
   - Follow-up: Have you noticed any challenges or drawbacks associated with IaC implementation?
   - Additional: Do releases become more streamlined? Can you discuss how processes became more efficient?

Part 2: Prioritization of IaC Metrics – discuss metrics

1. Do you currently have established metrics you use to monitor the efficiency of the IaC implementation at Jumia?
2. Are there any other metrics you believe should be considered? Why should they be considered

3. Here is a list of metrics (link) identified in previous studies. How would you prioritize these metrics relative to each other in evaluating the efficiency of IaC implementations?
4. What factors influence your prioritization of specific IaC metrics?
5. Why did you choose these factors?

Part 3: Metric Measurement and Tracking

1. How do you collect data for the metrics you use to track IaC's efficiency?
   - Follow-up: Are there any challenges you encounter in collecting or managing this data?
2. What tools or platforms do you use to monitor and analyze IaC metrics?
3. How do you ensure the accuracy and reliability of your metric data?
4. How do you address challenges such as data quality, data latency, and data integrity in your IaC metrics?
5. How good are these metrics in tracking efficiency as it is defined at Jumia?
6. What is the impact from measuring all of these metrics? Do you act upon them?

Part 4: Impact on Organizational Efficiency

1. Can you give me some examples of how IaC has benefited áreas outside DevOps processes?
   - Additional: Can you comment on developer satisfaction with the introduction of IaC?
   - Additional: Can you comment on the devops team satisfaction with the introduction of IaC?
2. How has IaC impacted Jumia's ability to deliver value to customers?
3. Can you share specific examples where IaC metrics have helped optimize the infrastructure and operations?

5.4 Data Collection

To gain the knowledge base required to answer the research questions I employed semi-structured interviews using a timeglass model.. Interviews were conducted with individuals directly involved in IaC development,and its management.
To that end, the sample base was specifically structured to capture a diverse range of perspectives within Jumia's IaC implementation. I focused on interviewing people with the following roles, and their corresponding significance to this case study:

- **Lead of Infrastructure**: This individual oversees the whole infrastructure at Jumia, and has input both on implementation and management of IaC. Their viewpoint will be high-level, giving insight on how IaC alligns with Jumia's broader goals and future strategy
- **Team Lead**: This individual plays a role in coordinating and guiding the development teams. Their perspective will be lower-level, providing insight into team dynamics, challenges and benefits faced during the adoption of IaC tools and practices
- **SRE/ Team Lead**: Serving as both SRE and team lead, this individual has first-hand experience with implementing and managing IaC, offering insight into technical challenges and benefits, while also providing insight into team dynamics
- **Site Reliability Engineer(SRE)**: This individual's day-to-day activities revolve around IaC. They provide insights into technical aspects, operational challenges, and the day-to-day impact of IaC on Jumia's developers and infrastructure.
- **Full Stack Developer(Interacts with IaC)**: This individual's main tasks are not IaC related, but they directly interact with IaC tools and SREs. They provide insight into the usability, effectiveness and areas of improvement in IaC workflows, and interactions.

All interviews were conducted between February and March 2024 using Google Meet and Microsoft Teams. The interviews lasted between 25 and 56 minutes, with a median duration of 40 minutes, totaling 5 hours and 20 minutes. For recording, I utilized a tool called "tldv", and for transcription, I used "Laxis". After the automated transcription, each interview was manually checked, with necessary corrections made. Subsequently, interviews were anonymized, and participants will be referred to by their code names.

Table 5.1 presents each interviewee by code, instead of name, their job title, and the duration of the interview.

Interviewee Code	Job Title	Years at Jumia	Interview duration
J1	Lead of Infrastructure	4	49 minutes
J2	Team Lead	4	36 minutes
J3	SRE/Team Lead	3	25 minutes
J4	Site Reliability Engineer	7	38 minutes
J5	Site Reliability Engineer	2	56 minutes
J6	Site Reliability Engineer	2	40 minutes
J7	Site Reliability Engineer	9	50 minutes
J8	Full Stack Developer	8	26 minutes

Table 5.1: The job title of each interviewee, along with the duration of the interview

5.4.1 Data Triangulation

In order to limit the effects of one interpretation of one single data source, and to increase the precision and validity of the research, data triangulation is required. [42] Runeson et al. remarks that there is an obvious need for data triangulation when dealing primarily with qualitative data.
There are four types of triangulation:

- Data Triangulation - using more than one data source, or collecting the same data at different occasions. [42]
- Observer Triangulation - using more than one observer in the study [42]
- Methodological Triangulation - combining different types of data collection methods [42]
- Theory Triangulation - using alternative theories or viewpoints [42]

To mitigate bias and one-sided conclusions, this case study will employ data triangulation by conducting interviews with multiple participants who hold various roles and levels of influence on infrastructure and IaC at Jumia. (as presented intable 5.1)
Data triangulation was done through cross-checking the information from the participants, as inconsistencies or variations in responses across different roles could reveal potential biases or areas needing further exploration. Moreso, by combining insights from various levels and areas within Jumia's IaC ecosystem paints a more complete picture of the implementation's impact and challenges.

5.5 Sampling strategy

Due to the limited timeframe of the study, the data sampling technique that would yield the best samples had to be chosen. To this end, I used non-probabilistic sampling, where people are deliberately selected for their specific characteristics withing the population. To note, while non-probabilistic sampling can be representative of the brother population, the goal of non-probabilistic sampling is to find accessible information-rich contexts from which the researcher can learn about the topic at study. [5]

Convenience sampling

Convenience sampling involves selecting samples based on availability or expedience. [5] In this case study, convenience sampling was employed to select participants who were directly involved with IaC on a daily basis at Jumia. These individuals were chosen on the basis of them being nearby, available or otherwise easy to study. [5]
Despite potential limitations related to generalizability [5] , convenience sampling is still a popular method, and in this case, was chosen for its speed and cost effectiveness.
This method yielded 5 out of 8 participants (62%)

Porposive sampling

Purposive sampling involves selecting samples according to a specific strategy, rather than at random. [5] This method allows the inclusion of participants that are highly relevant to the study, and exercise expert judgement.
In this case study, purposive sampling, specifically "homogenous sampling" was used to recruit participants overseeing decisions regarding IaC, and shaping strategies and policies related to IaC adoption and tools at Jumia.
Homogenous sampling is in direct contract to the "heterogenous sampling" approach where "the key dimensions of variations are identified and then cases are selected that differ from each other as much as possible.".[48]. In this case, I was purposefully looking for people that have power over decisions, policies and strategies related to IaC.

This method yielded 3 out of 8 participants (38%)

5.6 Ethics

Ritchie et al [40] remarks that ethical considerations are raised in any research study, but they are particularly relevant in qualitative research, due to its in-depth and unstrcutured nature. Qualitative research can raise issues that are not always anticipated, meaning that ethical considerations have a particular importance in these type of studies. As such, she poses four ethics points to be considered: Informed consent, anonymity and confidentiality, protecting participants from harm, and protecting researchers from harm.

- **Informed consent**: I have informed the participants about the study's purpose, the research team, how their data would be used, and the time required for their participation. I also informed them that participation is voluntary and they can withdraw at any point during the interview.
- **Anonymity and confidentiality**: Since the interview also delves into sensitive interpersonal relations, all participants were informed that their interviews will be anonymized, and their explicit consent was required for recording.
- **Protecting participants from harm**: Given the sensitive nature of the topic, participants were assured they could decline to answer any questions they felt uncomfortable with.
- **Protecting researchers from harm**: As the case study did not involve any potential harm to the researchers, no specific precautions were taken.

5.7 Data Analysis

5.7.1 Qualitative Data Analysis

To analyze the interviews, I chose to follow the guidelines established by Miles et al [33] and Saldana [44], as they enable a more deeper analysis.
Coding, effectively a data condensation task, is process where labels (codes) are assigned to data segments with the purpose of identifying recurring patterns and themes.As such, Codes can be considered prompts or triggers for a deeper reflection on the data's meaning. [33] Saldana divides coding into two main stages: First Cycle and Second Cycle.

1st Cycle

During the First Cycle, data segments from the interviews were assigned codes. The purpose of this phase is initially assign codes, that will be used in the next phase to identify recurring patterns.
I chose to use an inductive approach to coding, where codes emerge from the data itself, as opposed to a deductive approach that has predetermined codes. As defined by Miles et al [33], there are twenty-five different approaches or methods to do first cycle coding, each with a particular function or purpose.

After a first examination of the data, I decides to go with the following approached:

- **Descriptive Coding**: This type of coding summarizes data in a word or a short phrase, often as a noun. This type of coding allows insight into the social environment of participants. [33]
- **In Vivo Coding**: One of the most well-known coding methods, In Vivo coding uses short phrases or words, directly quoting the participants. [33]
- **Process Coding**: In Process Coding, codes are usually labeled as gerunds (words ending with "ing"). This type of coding implies actions are intertwined with the dynamics of time, such as things that emerge, change or become strategically implemented. [33]

Here is a breakdown behind he reasoning of choosing these specific approaches:

- **Descriptive Coding**: As noted by Miles et al. [33], Descriptive coding is useful for analyzing social environments. This alligns with my research focus, as I also explore how IaC impacts the environment of those that use it either directly or indirectly.
- **In Vivo Coding**: This method was chosen to directly refer to constructs or concepts related to IaC or Devops. [33] The participants often discussed these terms by explaining them, or giving specific examples (e.g., "drift detection","immutable infrastructure")
- **Process coding**: Miles et al. [33] suggest this type of coding is useful for extracting the participant's actions and consequences. This method is used to identify ongoing processes and actions related to IaC implementation. (e.g., resisting change, sharing knowledge)

2nd Cycle

The Second Cycle, also known as Pattern Coding, builds upon the initial coding done in the First Cycle, grouping these into small categories or themes. According to Miles et al.[33], Pattern Coding serves four key functions, three of which are particularly relevant for this thesis:

- Helps organize the initial codes from the first cycle into a smaller set of "analytic units".
- Starts the process of analysis during the data collection process
- Helps researcher create an evolving cognitive map for understanding local incidents and interaction
- Lays ground work for cross-case analysis by surfacing common themes, something that is not applicable for this thesis, as I am doing a single case study [33]

I started this process as soon as the transcripts were completed. This allowed me to monitor data saturation by continuously comparing data and emerging themes. By switching back and forth between data analysis and collection I was also able to better adapt the interview guide to further perfect data gathering.

Table 5.7.1 shows an example of pattern codes.

Pattern Code	Code	Example
Time investment	Migrating legacy infrastructure	*"So on when we migrated to AWS we also replicated the same setup and then when we later wanted to isolate staging from live, we hit the first roadblock. We had to recreate all the live set up again on the staging and it was a lot of trouble"* - J7 *"But sadly the live set up is always very hard to migrate and it was actually never applied ."* - J7 *"Yeah, so if you have resources already created, importing them it's a bit complicated but it's doable. "* -J1
	Initial setup time	*"Some setups on Terraform are easy and quick, but other setups require messing a lot with the variables, transforming variables from one kind to another, maps and objects and lists, and things like that. That is always a little messy in programming language, but HCL is a limited programming language and requires sometimes to deploy (to test)."* - J7 *"But I think in the beginning when Jumia was a startup, we were more pressured to deliver new things and sometimes to... I think it depends. But the initial effort sometimes is easier to set up something manually via AWS and we did work like that sometimes in the beginning."* - J4 *"the drawbacks can be your initial time investment... It's really just cost benefit. Are we going to build any more regions? Yes. Okay, well, then it makes sense. It's an investment one of it. If we're not going to, then maybe it makes sense to just swallow the cost of having someone manually sit there for 25 days straight and do a deployment"* - J6
Learning Curve	Immutable infrastructure concept	*"deploying the infrastructure that describes a Postgres database. Well, if I bumped the, if I changed the name of the database, it's going to actually spin up a brand new database, but none of the migration of the state to the new database is taken care of, for me."* - J6 *"For example, we had an incident with OpenSearch on AWS. We needed to replace the instance because the previous one got locked. Unfortunately, AWS didn't have enough nodes to automatically replace them as configured. We had to manually intervene and provision new nodes using AWS directly. This created a drift in our configuration. Later, when we tried to use Terraform to manage the cluster, it wanted to completely delete and recreate everything because it detected a difference in the configuration compared to the Terraform code. This could have been a big issue, potentially deleting and recreating all the OpenSearch nodes. Luckily, we caught it before any damage was done."* - J7
	Language issues	*"In Jumia you have multiple ways to achieve the same goal not all of them are as efficient. So we are also working a bit on on on making things a bit more segregate and in the end more portable. And yeah, it has some somewhat of a big learning curve, but I I think that the end results are are good."* - J3 *"the learning curve (is there) because the languages are very different from what the developers are used to"* - J8 *"one of the drawbacks could be that sometimes this code could be a bit specific for a given version of of either AWS or some specific it was component and could become outdated and not work so well so well. Sometimes we still have the felt some limitations of of infrastructure as code as well and we had to proceed manually."* -J4

5.7.2 Literature Search

To address the fourth research question (How do the challenges and benefits associated with IaC implementation, as experienced by Jumia, compare to the broader literature on the topic?) the results from the Pattern coding were analyzed against a literature dataset.

The literature dataset consisted of documents gathered through the MLR, along with an additional follow-up literature search aimed to identify studies that are either published after I conducted the initial literature search, or have not included into the MLR dataset.

Chapter 6

Findings - Case Study

In this section, the findings of the case study will be presented. These findings are derived from data analysis done on the semi-structured interviews. Recall that RQ3 seeks to understand how software practitioners expect to evaluate the efficiency of IaC implementations, while RQ4 aims to compare the challenges and benefits associated with IaC implementation at Jumia compare to the broader literature on the topic.

Although not essential for addressing RQ3 3, I made the choice to investigate the interviewees' perceptions of efficiency. This decision was driven by the aim of increasing alignment between the study's outcomes and existing literature. By comparing the metrics identified through thematic analysis with those outlined in the MLR, and by assessing whether practitioners' views on efficiency correspond to the definition provided in the MLR, we increase the homogeneity of the study. Furthermore, if the benefits and drawbacks reported by interviewees coincide with those documented in the literature, it also points to a higher overall level of validity and credibility of the findings.

The findings for each research question will be presented in a different section, to make a clear delimitation between the two.

6.1 RQ3

6.1.1 Theme: Metrics

Pattern Code: Lack of specific IaC metrics

Despite the growing adoption of IaC at Jumia, and the potential benefits, the interviews revealed a gap in how SRE's evaluate its efficiency. Specifically, none of the

interviewees mentioned the use of established metrics to measure the efficiency of IaC implementations.
This absence could be partially attributed to the reactive approach to Jumia's IaC practices. As J1 explains, Jumia created a Platform Engineering team in order to *" start to implemented the same standard across everything "*. This includes defining metrics for IaC.

> *"We are still, since this new team is really recent, we are still defining what will be the process of the team, what will be everything. But for sure that (there) will be metrics regarding infrastructure as code performance as well."* - J1

Simmilarly, J7 does acknowledge the lack of IaC specific metrics. *"No, directly from the infra as code I don't seem to have any metrics."*.
However, the absence of these metrics makes it hard to quantify the efficiency gains, or point to areas that need improvement. While some, like J8, rely on subjective perception , this approach lack the objectivity and granularity to make informed decisions about the IaC.

> *"We we don't measure it. We perceive the efficiency.*
> *We just don't know how much we are benefit benefiting from it ."* - J8

J7 gives an example of this sentiment in practice:

> *"We apply the Terraform ,know that it is fast, but we don't even measure that."* - J7

Fortunately, there is recognition of this gap and a willingness to address it. Both J1, J7 and J2 express an intention and hope that IaC specific metrics will be implemented in the future. *"Later on when we are more advanced, we will probably do that"* (J7) and *"...for sure that will be metrics regarding infrastructure as code performance as well"* (J1). *" ...we think that metrics related to infrastructure performance will certainly be a priority as we continue to evolve"* - J2
Nontheless, even though the benefits are clear, and there is a will to change from the interviewees, establishing a more data-driven decision making process remains a challenge for Jumia.

Subtheme:Infrastructure metrics

Pattern Code: Failure to correlate current metrics with IaC

Interestingly, after being presented with the list of metrics from 4.8, some interviewees, like J3 and J6 initially struggled to find a direct correlation.

> *"We do have some metrics (...)*
> *but they are not entirely tied to infrastructure as code."* - J3
> *"Some of these are, some of these don't seem related to me. "*- J6

J3 highlighted metrics like uptime, availability, and deployment frequency, but emphasized these focus on the broader DevOps cycle rather than IaC specifically: *"not all of them are entirely focused on infrastructure as code practices..."* - J3.
J7, while more receptive, reinforces this notion saying:

> *"There are some metrics here that are I think they are more focused for the development teams than the infrastructure guys to be honest."* - J7

Others on the other hand, did find a correlation, with J8 mentioning that this list of metrics is good at measuring the efficiency of IaC implementations.

> *" And then so overall you would say that all of these metrics would be quite good at measuring the efficiency of the infrastructure as code implementation?"* - Alexandru (Interviewer)
> *"I think so, and I never I've never read such a comprehensive list before, so I definitely yes."* - J8

Nontheless, even for metrics that could potentially be applied to IaC, a lack of active measurement is present.J7 acknowledges Jumia is measuring deployment times for Terraform, but it's not actively tracked :

> *"But in a quick view, we know about all these metrics but we don't actually actively measure the the those those changes related to the the infrastructure as code . We know the response time for the sites but not the how much time do we have to wait for a downtime to recreate things."* - J7

Pattern Code: Infrastructure-only metrics

While the interviewees acknowledge the importance of metrics, Jumia's approach currently focuses more on monitoring the health of the infrastructure, rather than evaluating the efficiency of its IaC.
J3 highlights their use of infrastructure-related metrics: *"We do have infrastructure related metrics... It's more like CPU usage, memory allocation, storage capacity monitoring..."* In the same tone, J6 adds:

> *"...like monitoring the infrastructure itself? Yes. Yeah, absolutely. Everything, everything you can measure we measured ."* - J6

These metrics provide good insights into the health of underlying resources like Kubernetes clusters and EC2s, but they don't directly assess how IaC code is performing.
Image 6.1 and 6.2 show a screenshot of Grafana, along with all the metrics they monitor for a specific K8S cluster.

This disconnect between infrastructure monitoring and IaC evaluation might suggest that rather than measuring the efficiency of their IaC implementations,

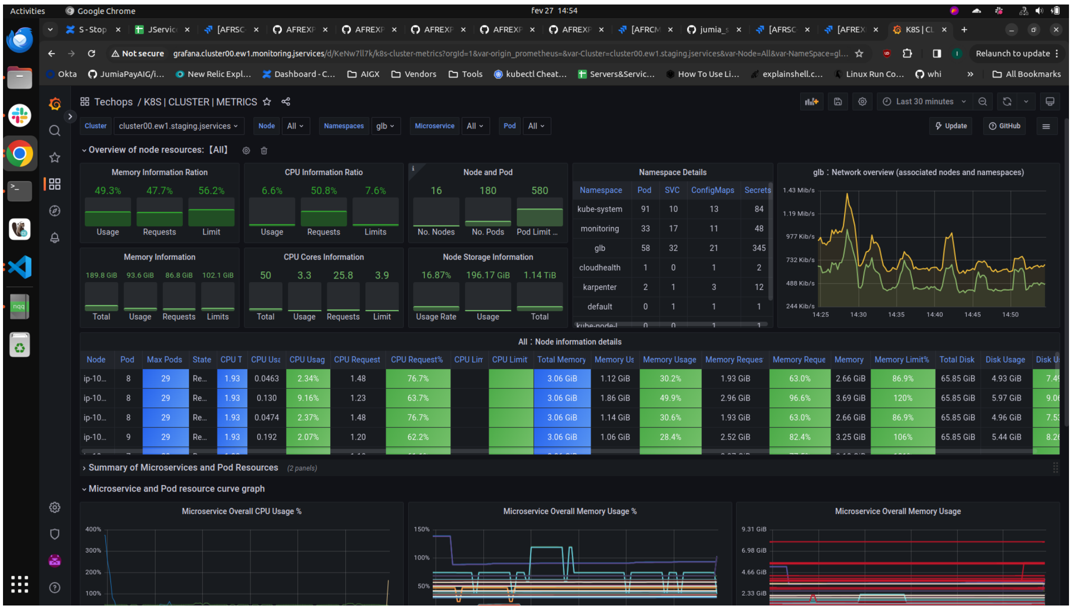

Figure 6.1: Grafana dashboard with metrics measured

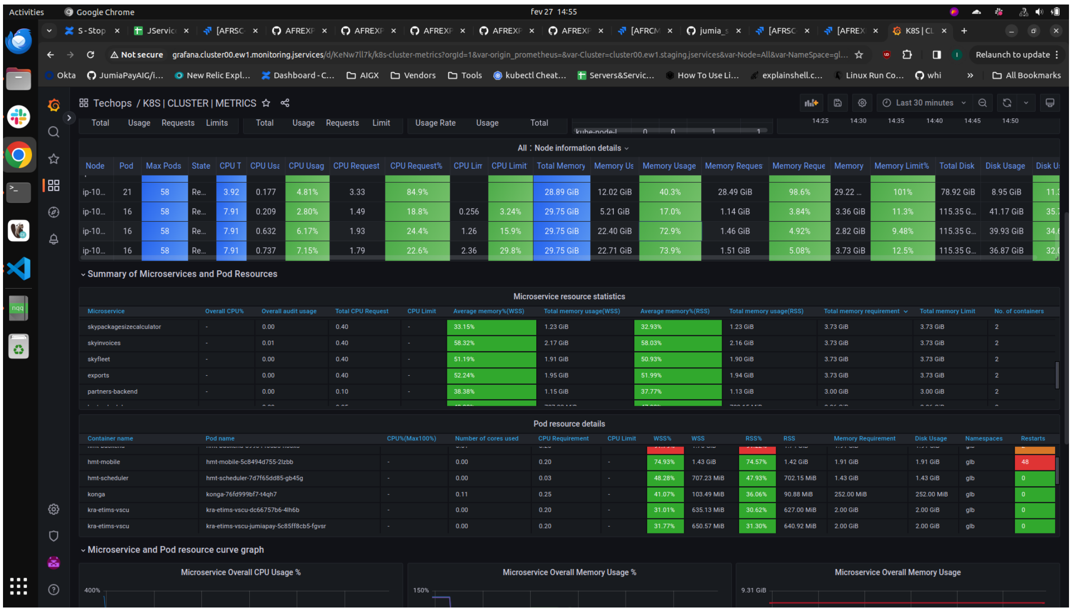

Figure 6.2: Grafana dashboard with metrics measured

Jumia focuses on identifying and resolving immediate infrastructure issues, a mark of a more pragmatic approach to their IaC implementation.
A reactive approach, like this one can be seen in J3's example of database storage alerts.

> *" So each time there are more products being uploaded to Jumia by the vendors, the databases that support the Jumia catalogue will get bigger and bigger and bigger. And if we reach a threshold of 90%, so usually we have an 80% alert that will go through our slack channels and the teams should already act on it. But if by some reason we don't act when we reach the 90% threshold we receive a call and then we should really act because it might take just a few hours for the database to become full and we might have an outage ."*- J3

This reactive approach focuses on ensuring the immediate health and availability of the infrastructure, which alligns with their business needs of maintaining uptime and preventing outages.
This approach allows them to address critical issues swiftly (such as database-caused outages), and ensures the infrastructure can support the dynamic demand on the platform. Nontheless, by not actively measuring efficiency related metrics, Jumia risks falling behind on the continuous improvement cycle, something that can result in possible missed opportunities for optimizing IaC practices, and inneficiencies.

Pattern Code: Efficiency metrics prioritization

So far, it's evident that Jumia does not currently track any specific metrics when discussing the efficiency of their IaC implementation. However, to gain insight into which metrics are considered most crucial, I asked all the participants for their opinions. The following list includes the metrics that at least half the participants deemed very important, with the exception of Deployment Time Duration,which is presented because it was the 5th most mentioned in the literature, but was only mentioned by 1 interviewee as most important.

An interesting observation is that these metrics align with the findings of the MLR. The main difference is the emphasis on Infrastructure Code Coverage, which was the most discussed and mentioned metric. This focus is possibly due to the current low IaC coverage at Jumia, with only 40% of the infrastructure being covered by code, as noted by J1.

Code: Infrastructure code coverage

Among the metrics discussed, Infrastructure as Code Coverage stood out as one of the most critical, despite not being present among the top five metrics commonly mentioned in literature. Its importance was highlighted by six out of eight interviewees.
J1, the head of infrastructure, discussed the current state, and importance of IaC code coverage.

> *"I would say that we will have around 40% if we count all verticals"* - J1

He also mentioned that without a high coverage, the stability and reliability of the infrastructure remain questionable.

> *"So for me the number one would be infrastructure code coverage for sure. Because it will tell you if you have a good base for things to be stable in the future"* - J1

Low coverage suggests a reliance on manual processes, and a less than ideal foundation for scalability. In his view, a stagnant coverage metric implies a lack of progress in automating infrastructure management.

> *"Because it means that it's still being done manually, we still have the mentality of doing things manually"* - J1

This sentiment also resonated with others. J7 acknowledges that the current coverage is *"still a bit far from"* the desired 100%, but also emphasises the goal to increase it.
J5 also linked the high coverage to minimized manual interventions,a key benefit of IaC.

> *"I wanted to have the infrastructure coverage more where I can avoid the manual intervention like, for example, instance deployment, cluster creation whatever it may be"* - J5

Overall, the interviews revealed a clear consensus that an improved IaC coverage will lead to a better long-term stability and a higher efficiency of Jumia's infrastructure.

With only around 40% IaC coverage, the reason the employees at Jumia were so keen on this metric steps from Jumia's reliance on manual processes, that hinder scalability and automation efforts. With Jumia being present in over 11 countries, having easily reproducible IaC components, that are easily to deploy, manage and bug-fix is essential, both to ensure long term stability, and to minimize manual interveitions.

Code: Time to Restore Service

The second key metric that emerged as critical is Time to Restore Service. This was the second most mentioned both in the literature and in the interviews, highlighted by five interviewees.
J1 emphasized the importance of IaC for disaster recovery

> *"It's really important to have infrastructure as code for disaster recovery"* - J1

He believes that a well-structured IaC foundation, with a good coverage can significantly reduce recovery times compared to manual processes. However, he acknowledges the limitations of setting absolute time targets:

> *"So if you have a good coverage, if you do everything with Terraform and you know and you have everything well structured and a good foundation as well, the time to restore will be faster than without it. Of course you cannot say I want time to restore to be 10 minutes. When the resources take one hour to provision, you cannot read, you cannot go lower than this value."* - J1

J7 further reinforces this point by emphasising the important role of an efficient service restoration process:

> *"It's not something that we do every day, but it's very critical if we have a major incident and do we want to create everything."* - J7

The reason why the employees were so keen on choosing this metric stems from the nature of company. As an online e-commerce platform, Jumia relies on its always online reputation. If the vendors are not able to sell their products, or the shoppers are not able to buy products, they will turn to other companies that allow them. As such, the companies revenue, somewhat indirectly lies on this metric.

Code: Number of Drifts

The third most mentioned critical metric is the Number of Drifts. Despite not being in the top five list of most mentioned metrics in the literature, its criticality was underscored by half the interviewees. J1 sees the number of drifts as an indicator of the quality of IaC, as fewer drifts means less manual interventions.

> *"Number of drifts is important because it will tell you if your infrastructure as code is as good as you think"* - J1

J7 also stressed the importance of detecting drifts, to ensure that configurations stay as intended, this feeling also being echoed by J2:

> *"Number of drifts that one is I think is important. If things that we detected are different from what we wanted, I think is very important."* - J7

> *"The number of drifts and ratio of drifted to total resources metrics could help us ensure that our infrastructure configurations stay consistent and aligned with what we want"* - J2

J5 also highlighted this need to minimize drifts, especially in a production setting:

> *"The number of drifts should still be very low when it comes to the production environment"* - J5

Uncontrolled drift can lead to inconsistencies that require a lot of work to fix. Jumia's experience with infrastructure drift, as described by J8, exemplifies the negative impact on cost and efficiency.

> *"Yeah , I I kind of remember that at some point we were using EQS instances from AWS and the cluster serving an application was having different hardware instance classes and for the worst meaning that we were paying more for hardware that we didn't need."* - J8

Since Jumia is an e-commerce platform present in over 11 countries, the employees found this metric very relatable to their issues. On one side, customers expect a very similar shopping experience, whichever country they may shop in, and on the other side, the developers code, and deploy the same platform over all the countries.
Minimizing drifts maintains a stable and secure environment, reduces the risk on errors, and gives a seamless and reliable experience to the customers.

Code: Deployment Frequency

The next metric that emerged as critical to evaluate the efficiency of IaC was Deployment Frequency. It was the fourth most mentioned metric in the literature, and was also highlighted by half the interviewees as critical.
J1 gives an example into the context-dependent nature of Deployment Frequency. He mentioned that while it is generally important, it may not always be relevant for all IaC components:

> *"It will depend on the case, because with infrastructure as code I don't think it makes a lot of sense because if you create for example a VPC with reverse section score, you don't want that to change, so there's no need to measure the frequency of the deployment"* - J1

This suggests that while frequent deployments can indicate a more responsive infrastructure management, certain stable components may not require frequent changes.
Nontheless, Deployment Frequency remain top priority, as it often reflects the ability to quickly adapt and implement changes, including security patch deployments.

Jumia's e-commerce platform benefits from a high deployment frequency to swiftly adapt to market demands and implement feature enhancements. For instance, during peak shopping seasons like Black Friday or holiday sales, the ability to deploy updates rapidly allows Jumia to optimize user experience, manage increased traffic loads, and capitalize on sales opportunities.

Code: Lead Time for Changes

Another metric that emerged as most critical from the interviews is Lead Time for Changes. Mentioned by half the interviewees as critical, this metric measures the time taken to implement, test and deploy code changes. Lead time for changes was also the most mentioned metric in the MLR.
J1 emphasized the importance of tracking this metric, noting that:

> *"time to implement should be slow in the beginning... with the team improving and with the knowledge of teams, raising, we hope that implementation time goes a lot lower"* - J1

He highlighted that as the team gains experience and knowledge, the lead time for changes should significantly decrease.
As such, the metric could also point toward the IaC maturity. A more standardised infrastructure as code implementation increases knowledge sharing, which in turns results in a quicker lead time for changes. J2 echoes the same sentiment, linking a lower lead time for changes to a faster problem resolution.

> *"A lower time to restore service means we fix problems faster, because of the Terraform"*- J2

This illustrates that the metric can also be used for tracking the time to fix bugs perhaps.

For Jumia's e-commerce platform, minimizing lead time for changes is crucial to staying competitive in the fast-paced online retail market. With rapid shifts in customer preferences and market trends, the ability to quickly implement and deploy code changes directly impacts the platform's agility and responsiveness. For example, reducing lead time enables Jumia to quickly roll out new features, optimize user experience, and address emerging issues, ensuring that the platform remains relevant and appealing to customers.

Code: Change Failure Rate

Another metric that stands out as very important is Change Failure Rate. It was the third most mentioned metric in the literature, and was identified as critical to measure the efficiency of IaC implementation by half of the interviewees.

J1 emphasized the importance of ensuring changes are thoroughly tested in development before being applied to production:

> *"We don't want to apply changes that will fail and weren't tested properly in the development environment. So we need to take really good care of production"* - J1

This focus on stability also reflects as a benefit of IaC, namely the ability to automate repeatable deployments (even in different regions) with a reduced risk of error. When implemented correctly, IaC minimizes the chance of configuration drift, or human error that can lead to deployment failures.
He also discusses the importance of staging environments:

> *"It's really important that you have a good process of implementing changes... usually you have at least one previous environment, so you will have the opportunity to do things right"* - J1

J1 highlighted that frequent failures often indicate discrepancies between environments, which should ideally be stable and consistent:

> *"You need to understand what you are changing on staging and if you have a lot of failures, it's because your environments are different. You are diverging both environments that should be more or less equal stable because you want to make changes and you. You want to know the output of your changes or the expected behavior."*- J1

J5 also pointed out the need for thorough testing across environments to minimize failure rates:

> *"Usually, once we test and implement the code under any environments, the percentage will usually be very low"* - J5

While Jumia doesn't explicitly measure change failure rate, their focus on preventing failed deployments through testing and consistent environments aligns with the principles behind this metric. This focus on stability suggests Jumia prioritizes reliability as a key measure of IaC implementation efficiency.
Moreso, this metric is important to Jumia in order to ensure a seamless customer experience and business continuity. For instance, imagine a scenario where a new feature deployment fails due to inadequate testing, resulting in website downtime during peak shopping hours. This not only disrupts customer transactions but also damages Jumia's reputation and revenue.

Code: Deployment Time Duration

During the MLR, Deployment Time Duration came up as the fifth most mentioned metrics when discussing efficiency of IaC implementations. While it was the fifth most mentioned metric, only one interviewee highlighted it as of critical importance. The reason behind this being that deployment time, at Jumia, is depended on AWS's provisioning speed, not the efficiency of the IaC code. J1 gives an example of this possible bias from the cloud provider. He elaborated that the time AWS takes to create resources, such as an EKS cluster, is fixed:

> *"If you have a pipeline to run the creation of an EKS cluster, it will take 20 minutes to create because it's the time that AWS takes to have that results ready"* - J1

Hence, there is little that can be done to improve this duration from the perspective of infrastructure coding.

6.1.2 Theme: Perception of efficiency

The second theme delves into how the interveiweees perceive the efficiency of IaC implementations. As explored in the next chapter, this perception is closely tied to the benefits seen from adopting IaC. Interviewees highlighted two main areas of efficiency: cost reduction and time saving. Cost reduction was seen through infrastructure optimization and identifying unnecessary expenses, leading to a more efficient resource usage.
Time saving was another key aspect of efficiency, shown both in the speed of infrastructure deployment time, and time savings in configuration management, (also referred as a reduction in "click-ops"), and a reduction in manual workload.

Pattern Code: Cost reduction

When addressing how the software practitioners at Jumia define the efficiency of IaC implementations, a common theme is the perception of efficiency through cost reduction. This includes both infrastructure expenses, and operational expenses. J3 highlights the impact of IaC on operational expenses, explaining how IaC enables rapid provisioning of new servers for applications, leading to quicker scalability either up or down, and implicitly money savings.

> *"With infrastructure as code, we were able to quickly provision new servers for new applications to run on top of, and we are able to do this in a more scalable way"* - J3

J7 gives a practical example of cost savings achieved through IaC by preventing wasted resources. He mentions finding an unused database that had been incurring costs:

> *"Even a few days ago I found a database in a zone that was created for a test and was never deleted. It was a small database but we were paying for it for a long time, it was useless"* - J7

This perspective on efficiency aligns with the benefits mentioned in the next chapter. Through standardisation, easier scalability and transparency, IaC can help teams avoid human errors, improve resource allocation and ultimately reduce the cost overhead.

Jumia sees efficiency in IaC as a cost reduction, crucial for a company struggling financially that has yet to turn a profit. By quickly leveraging IaC, they can optimize operational expenses, such as server provisioning, leading to quicker scalability, and implicitly cost reduction. Additionally, IaC helps and identify wasteful resources, such as usused databases or EC2s. This alligns with Jumia's future goal of achieving financial sustainability.

Pattern Code: Time reduction

Another significant theme that emerged through the interviews is the perception of efficiency as time reduction. This either comes through a decrease in cycle time, or through the reduction in the manual workload.
J1 discussed that he views efficiency through the decrease in lead time for changes, particularly in resolving tickets received by the development team:

> *"In our case, it was the time for us to solve the tickets received by the development team... tasks that we were doing were really easy to automate, so they were not automated and we were losing too much time in each ticket"* - J1

This means that SRE's could take on a higher number of tickets in the same amount of time, effectively increasing their efficiency.
J8 emphasizes that he sees efficiency in IaC as the reduction of manual tasks, highlighting the efficiency of making changes directly in code:

> *"As opposed to having to go to a web console... it's much easier for a person to efficiently change... it's quite simple to go to the code, change it to whatever you want, commit, push, and things will just happen"* - J8

J6 and J3 echo this sentiment, further pointing that:

> *"the highest level of maturity and infrastructures code is it's pretty easy to say that - like zero click ops"*- J6

This perception of efficiency also alligns well with the benefits mentioned in the next chapter, as IaC not only accelerates deployments, but also reduces manual workload, leading to more efficient handling of tasks, and simplicity a faster time to market for new features. This is crucial for Jumia as they can more quickly adapt to customer and market demands, tasks that tie in with the company's goal of providing a seamless shopping experience for its customers while staying ahead of competitors.

RQ3 Conclusion: The evaluation of IaC efficiency at Jumia reveals a perspective shaped by their operational priorities. While efficiency metrics are not consistent with the MLR's claims, the emphasis lies on reactive measures related to immediate infrastructure health and service availability. This reactive approach, evident in their hardware-related metrics, and response procedures toward them underscores their focus on resolving critical issues promptly to ensure uninterrupted service, a very important goal for an e-commerce platform.

However, the absence of proactive efficiency metrics might lead to possible missed opportunities for continuous improvement and cost optimization within their infrastructure.
Nonetheless, when asked about the most critical metrics for evaluating IaC efficiency, the interviewees mentioned six of them: Infrastructure Code Coverage, Time to Restore Service, Number of Drifts, Deployment Frequency, Lead Time for Changes and Change Failure Rate, each having an impact on Jumia's motivation to maintain high Service Level Agreements, such as availability or loading times, and staying ahead of the competition.

Moreover, when asked about their perception of efficiency within IaC implementations at Jumia, the interviewee's responses revolved around achieving cost reduction and time savings.

6.2 RQ4

To answer the fourth research question, as mentioned in chapter 5.7 I will first present the benefits and the challenges of IaC implementation at Jumia, afterward will present the findings of the literature search, and will finish by correlating the two, findings commonalities and discrepancies.

6.2.1 Theme: Benefits of IaC Implementation

Subtheme: Standardization

One key theme centers around the reduction in time to market. Overall, through all of the things IaC does, Jumia gets a shorter time to market.

Pattern Code: Improved reusability of infrastructure code

A core benefit of IaC that was relevant in all interviews is its ability to promote reusability of IaC configurations. This directly leads to faster deployments, improved consistency and inherently an increased efficiency and throughput of development teams. J3 emphasizes this advantage, *"the time to market from our perspec-*

tive is much reduced with infrastructure as code...we spend much less time provisioning the infrastructure." By thinking of infrastructure as code, teams eliminate much of the need for manual setup for each new deployment.
Further amplifying this benefit is the possibility of creating consistent deployments across multiple environments, which helps lower the risk of errors and configurations drifts. Moreso, by reusing code, IaC also enforces a "standard" upon the code that is written. J7 notes that *"applying the terraform allows us to make sure that the setup is the same on the both places."* when discussing multiple environments.

Reusability also helps scale the infrastructure easier. J7 again notes this:

> *"So the next zone that we'll create will grab the same terraform and apply the the same terraform to a different zone and have exactly the same set up as we have on Ireland . So that will also allow us to multiply the the set up to four different zones without any almost any extra work."* - J7

J1, the head of infrastructure reinforces this point by describing how IaC streamlines handling new requests:

> *"every new request if it's on interest of risk code it will be a matter of a new PR that needs to be created...or if it's something new for the infrastructure is just seeing how we can do it in Terraform, implementing it one time and all the sub requests apart from that one will be really fast to do."* - J1

By creating reusable components, IaC allows teams to handle "boilerplate"tasks much easier.
The concept of reusable components is also mentioned by J1 and J4, discussing Jumia's centralized repository: *"We created workspaces per environment, so the same module will be used, we will just change the values...So it's just to to make sure that everything is using the same thing across all the environments."*. J4 also gives an example of reusability in action:

> *"For example, I...was working on the making the internal communication between two different Jumia VPCS Jumia Pay and Jumia Logistics...most of my work was just looking at configurations that were already in place...For other similar configurations I just had to add on the correct places on basically a yaml file..."* - J4

J5 gives a similar example: *"Means I can say it's like in Terraform we can write the modules and make the modules reusable so that I can just change my variables."* By leveraging existing IaC code and components they were able to implement a new configuration with a low amount of effort.
The reusability of infrastructure code has a significant impact for Jumia, as they can achieve faster deployments, and improved consistency across environments, and the 11 countries they are present in. This not only quickens time to market for new features across countries, but also facilitates the expansion into new regions or coutries.

Pattern Code: Increased transparency

Beyond just reusability, IaC also promotes transparency, which leads to better collaborating teams. The IaC codes acts as a shared document, allowing everyone to understand what the infrastructure should be and do. J3 highlights this:

> *"Because for instance, let's say the developer that we have that we are going to have some big campaigns on Jumia and we need to scale our infrastructure. We with infrastructure as code, we can do this as code, truly transparent to everyone"* - J3

The transparency also fosters collaboration and knowledge sharing. J1 describes how IaC code is used as a reference point: *"you can see that people read the Terraform code that you have and understand what they need to change."* This improves handoffs between teams when interacting with shared infrastructure, a factor also mentioned by J5: *"Also very helpful for the people for the organizations to maintain. If some engineer leaves and there should not be any dependency with the engineers"*. Finally, IaC also promotes consistency across projects and environments through transparency. Engineers can thus rely on familiar patterns and configurations.

> *" So it's easier for one SRE from 1 project to start to messing with another project because the set up is closer. In the past we all the the projects were using totally different solutions. Right now we try to to set up everything on the the same way, using the the cloud as as much as possible and some standard so on the on the setup so it's easier."* - J7

Increased transparency is also a significant advantage for Jumia, a company whose median tenure [28] is less than half the average of other companies [52], as it increases knowledge sharing and increases collaboration across employees, ensuring continuity even when personnel change occurs.

Pattern Code: Monitorization

One benefit mentioned by J1, the head of infrastructure is the impact that IaC brings to monitorization.

> *"A monitoring as well, because once you have everything standardized, it's easier to create the same standards for monitoring the same alerts for everything and be comfortable managing different infrastructure, even if they are different verticals inside the company."* - J1

By ensuring standardisation through IaC, developers can remove potential inconsistencies that complicate monitoring. This allows for the creation of monitoring templates and alerts that can be applied infrastructure-wide.
J5 further pushes this benefit, as we can now use *"continuous monitoring tools during deployment to make deployments more robust while also gathering helpful feedback"*.

In Jumia's case, monitoring plays an important role in their reactive approach to IaC. Through hardware related metrics they keep track of the infrastructure's health and availability. Moreso, IaC also allows the monitorization of efficiency related metrics, such as the ones presented in RQ3.

Pattern Code: Enhanced security

Mentioned by only one interviewee, IaC also improves security through its control over secrets and credentials. By managing these sensitive resources through IaC and storing them in Jenkins, we eliminate the risk of unauthorized access.

> *"But to putting like this on the on Jenkin or RunDeck allow us to do operations with secrets and passwords and access they don't have and they can change some parameters some configurations running like that."* - J7

Furthermore, IaC also promotes a "clean as you go" approach to infrastructure. As explained by J7, *"So if we remove the resource from the IaC code, it will be automatically removed, so it's easier to to clean up and give us more security with the change ."*. This ensures that outdated resources are no longer staying in the environment and creating security vulnerabilities.
The enhanced security brought on through the implementation of IaC is another desired benefit by Jumia. Safeguarding sensitive customer data and transactional information is imperative for an e-commerce platform. Moreso, by managing secrets and credentials through IaC, the risk of unauthorized access is mitigated.

Pattern Code: Facilitation of continuous improvement

Beyond automation, one interviewee, the head of infrastructure also discusses how IaC fosters a culture of continuous improvement within the SRE teams.
Since the code serves as a form of documentation, senior members can give responsibilities to the junior members while still being able to monitor what they are doing. This fosters a more collaborative environment, where knowledge is more easily shared across team members.

> *"Also it's much easier to for the senior guys to to give responsibility to to the more junior team. So we it's easier to monitor what they are doing. It's easier to control and it's easier also for the team knowledge to grow with this and it's faster to bring infrastructure up."* - J1

IaC also promotes continuous improvement through standardisation. By coding using the same standards, in an open environment, teams can identify and implement best practices. This also create a cycle of continuous optimization, as new learnings further improve the code base. In J1's words:]

> *"we are bringing value because we we we have different verticals with different needs and we can with the proper standard and everything being equal or as much as they can be equal, we can pick up on the best solutions and create the, it's always improvement.It's always like creating a new revision of your infrastructure, that's that's also I think a value for for the company."* - J1

This is desireable by Jumia, because it increases their agility, crucial in the fast paced e-commerce environment. Through continuous improvement, Jumia can keep up with emerging technologies trends, and security issues, staying relevant for customers, regardless of their geographical location.

Pattern Code: Standardized toolset across development teams

Interviews also revealed advantages with IaC's ability to use a standard toolset across development teams. By using a common language, like Terraform, developers can more easily understand and start contributing to existing infrastructure configurations, eliminating the need for learning more languages. As J1 notes, *"everyone feels at home currently because verticals are starting to implement things. So when people need to see other Terraform or other repositories with Terraform code, they feel right at home."*.
J4 expands on this benefit by also mentioning the integration of IaC (and Terraform) with existing developer workflows.

> *"one of the big advantages I think of infrastructure as code is also that you can basically use all of the other tools that the developers use for, for the development of of code review, like pull requests, like peer reviewing, like testing, planning."* - J4

In the same train of through as transparency, having a standardised toolset across development teams is important for Jumia, as employees that join can more easily pick up on existing projects, eliminating the need for learning new languages or tools.

Pattern Code: Improved scalability

The ability to streamline infrastructure provisioning and scalability was also mentioned as one of the benefits during the interviews. J3 highlights the speed and ease of provisioning new resources, as IaC eliminates most manual steps that come with configuration and deployment.

> *" with infrastructure as code, we were able to quickly provision new servers for new applications to run on top of, and we are able to do this in a more scalable way."* - J3

This automation also enables easier scaling, as J7 mentions: "*And don't forget IaC also allows to quickly if we need to change the instance for something or because of load or something like that, we can do quickly change most of the those setups .*". IaC code can easily be modified to change resource allocation, which is visually represented just as a number.

The impact this also extend beyond provisioning, as J7 emphasizes the reduction in deployment time:

> *"So when in the past we took several days or weeks to do a task, right now we can update or change some service in seconds, minutes or hours only very big requests."* - J7

This agility allows Jumia to respond to fluctuations in demand much quicker, maintaining continuous service delivery, crucial for an e-commerce platform, especially during sales periods like Black Friday, or specific holiday where demands are at all time highs.

Subtheme: Disaster Recovery

Limitation of configuration drift
One of the most important benefits of IaC lies in its handling of configuration drift. J8 emphasises this being an issue before Jumia adopted IaC: *"Number of drifts, instances where the production environment deviates from its desired configuration, quite an important one. In fact we suffered from this in the beginning when we have no infrastructure as code and it we have a full mix of things in production where they should be fully standardized"*.
These drifts can have very big consequences, not only software wise, but also monetary.

> *"Yeah, I kind of remember that at some point we were using EQS instances from AWS and the cluster serving an application was having different hardware instance classes and for the worst meaning that we were paying more for hardware that we didn't need."* - J8

IaC helps prevent drift through its "immutable infrastructure" principle. Whenever someone applies Terraform, the software will scan over all the currently defined infrastructure, notify of possible drift and allow developers to correct them. This does have a downside discussed later on, in disadvantages related to IaC adoption, especially when handling stateful infrastructure.
This sentiment was echoed by both J4 as IaC allows for *"tracking and I would say the real state of the infrastructure"*. J7 also provides a real example:

> *"For instance in the past we had some configurations that were applied by Ansible and then someone would change it by hand... and there was a time that we broke the server for a couple of minutes."* - J7

Immutable infrastructure brought on by IaC also simplifies disaster recovery, by simplifying the rollback process, for stateless infrastructure at least. This is exemplified by J5:

> *"Whatever the changes which happened it will be like I apply Terraform code, and if have something has been changed and I would just wanted to revert it back that can be changed immediately without affecting the other resources."* - J5

A simplified rollback process also leads to a faster spin-up of downed infrastructure, a sentiment also echoed by J1, *"if you do everything with Terraform and you know and you have everything well structured and a good foundation as well, the time to restore will be faster than without it."*
In line with Jumia's desire to uphold its reputation and customer trust, limiting drift configuration is vital in order to maintain consistency and stability across the multiple environments, to keep availability at a maximum. Aditionally, by mitigating drift, in case of emergencies, the infrastructure can be quickly rolled back to a stable state, crucial in order to maintain availability.

Pattern Code: Fostering DevOps culture

The interviews also revealed that another of its benefits is its role in promoting a DevOps culture. IaC, by promoting collaboration between the development and ops teams, and between the different silos, tries to break them down, and increases the level of ownership of the infrastructure.

By enabling developers to manage infrastructure through code, IaC reduces bottlenecks caused by the smaller OPS team, and empowers them to take more ownership of their deployments.

> *"we were able to offload a lot of the work... Developers then became a bit more autonomous."* - J8

J1, the head of infrastructure emphasizes the importance of a *"proper DevOps culture"*, and *"giving the proper ownership to developers"*. He views IaC as a critical tool of pushing this culture where developers assume greater responsibilities in infrastructure, along with their main focus on application development.

> *"So we want to implement a proper DevOps culture and want to to give the proper ownership to the developers. So we will start shift left some responsibilities that should always been responsibility of the development teams, but we want to do it properly. So once we shift, they will have all the tools to do it, they will understand the responsibilities and it will only be possible with infrastructure as code."* - J1

J4, an SRE also agrees with this statement that that IaC *"can benefit... the developers to also contribute to the infrastructure"*. By defining infrastructure as code, the layer of manual processes, and knowledge harnessing without sharing is removed. This

transparency, as J6 suggests allows the developers to move beyond point-and-click interfaces and gain a deeper understanding of the *"reality of what they're actually building"*.
Fostering a DevOps culture is important for Jumia as it promotes collaboration,agility and innovation across development teams. By breaking down silos and encouraging cross-functional collaboration, a faster delivery of features is promoted, and infrastructure reliability can be increased, all required to quickly respond to market changes and customer needs.

6.2.2 Theme: Challenges of IaC Implementation

Two main pattern codes emerge from the study: Time Investment and Learning Curve

Pattern Code: Time Investment

The most common response from participants when discussing challenges was the large time investment required to implement IaC. This drawback came apparent at three stages during the lifetime of a IaC project. Either when migrating legacy infrastructure that is not deployed using IaC, when creating new projects from scratch, or when provisioning already created projects.
Code: Migrating legacy infrastructure
Migrating legacy infrastructure to IaC can be a complex task. This is highlighted by J7, when Jumia first switched from legacy to AWS.

> *"So on when we migrated to AWS we also replicated the same setup and then when we later wanted to isolate staging from live, we hit the first roadblock. We had to recreate all the live set up again on the staging and it was a lot of trouble"* - J7

The complexity of replicating a live environment with all its existing dependencies and configurations can make IaC adoption a very slow process, and it can even hinder it completely. *"But sadly the live set up is always very hard to migrate and it was actually never applied ."* - J7.
While technically feasible, importing existing resources into IaC is a complicated task, as J1 acknowledges *"Yeah, so if you have resources already created, importing them it's a bit complicated but it's doable. "*, but it requires a deeper level of knowledge. Importing resources can lead to unintended consequences, and as such, there is a bottleneck created as it requires senior engineers to do it.

> *"But that (importing resources) requires as well a lot of knowledge and only usually the senior guys can do this properly without breaking anything,... The precedence of the resources that you import are really important because if you import in the wrong order, it can trigger the infrastructure as code to recreate everything."* - J1

This is a downside for Jumia due to the fact that only 40% of its infrastructure is through IaC. Jumia may face challenges in achieving consistency and scalability across its entire infrastructure ecosystem.as the complexities involved in migrating legacy systems to IaC can result in prolonged transition periods, increased resource allocation, and potential disruptions to ongoing operations, all undesirable for a company focusing on high availability.

Code: Initial setup time consumption

Even for new projects, the initial setup phase involving IaC configuration and environment preparation can be time-intensive. One participant mentioned the challenges associated with complex Terraform configurations, particularly working with variables and data structures.

> *"Some setups on Terraform are easy and quick, but other setups require messing a lot with the variables, transforming variables from one kind to another, maps and objects and lists, and things like that. That is always a little messy in programming language, but HCL is a limited programming language and requires sometimes to deploy (to test)."* - J7

This challenge was often more of an issue in the start-up phase of Jumia, as J4 mentions:

> *"But I think in the beginning when Jumia was a startup, we were more pressured to deliver new things and sometimes to... I think it depends. But the initial effort sometimes is easier to set up something manually via AWS and we did work like that sometimes in the beginning."* - J4

This need for rapid delivery could push teams toward manual configuration options, an antipattern for IaC. J6 also emphases the importance of considering the cost-benefit tradeoff.

> *"the drawbacks can be your initial time investment... It's really just cost benefit. Are we going to build any more regions? Yes. Okay, well, then it makes sense. It's an investment one of it. If we're not going to, then maybe it makes sense to just swallow the cost of having someone manually sit there for 25 days straight and do a deployment"* - J6

For smaller, one-time deployments the initial investment in IaC might not be reasonable. But, for ongoing deployments, or geographically distributed architecture like Jumia's, the long term benefits of automation and repeatability more than outweigh the initial investment.

Code: Learning Curve

The second most common response when discussing challenges was the Learning curve associated with IaC. This can be seen in both the different programming paradigm (declarative) that the developers are not used to, the concept of immutable infrastructure, that oldstyle cloud engineers were not used to, all leading to somewhat of a resistance to change from the developers.
The reason these downsides are significant for Jumia is through the significant training required for employees unfamiliar with IaC. Moreso, even though not very prelevant, resistance to change from developers can slow down the integration of IaC practices, impacting the ability to maintain agility and thus competitiveness in the e-commerce market. Moreso, by having such a high turnover rate, retraining employees can further take up the available resources.

Code: Immutable infrastructure concept

Immutable infrastructure, a core concept of IaC promotes consistency and security by treating resources as disposable. However, interviews showed challenges associated with its inflexibility and overhead of managing stateful, existing configurations, or unexpected situations. J7's experience with OpenSearch on AWS exemplifies this inflexibility. Manual intervention to replace a locked instance created configuration drift. When Terraform, enforcing the Immutable Infrastructure principle, tried to recreate the entire cluster, it could have resulted in data loss.

> *"For example, we had an incident with OpenSearch on AWS. We needed to replace the instance because the previous one got locked. Unfortunately, AWS didn't have enough nodes to automatically replace them as configured. We had to manually intervene and provision new nodes using AWS directly. This created a drift in our configuration. Later, when we tried to use Terraform to manage the cluster, it wanted to completely delete and recreate everything because it detected a difference in the configuration compared to the Terraform code. This could have been a big issue, potentially deleting and recreating all the OpenSearch nodes. Luckily, we caught it before any damage was done."* - J7

The overhead of managing existing configations with IaC further highlights this issue. For example, J6 had issues with a trivial name change of a stateful Postgres database.

> *..."deploying the infrastructure that describes a Postgres database. Well, if I bumped the, if I changed the name of the database, it's going to actually spin up a brand new database, but none of the migration of the state to the new database is taken care of, for me."* - J6

These challenges show a trade-off between the benefits of immutability and the need for some degree of flexibility when managing infrastructure. J1, the head of infrastructure put emphasis on this - immutability is a different way of thinking, and developers should be aware of it.

> *" Infrastructure as code should be easy, easily readable, should not be too complex and if you are doing complex things you are doing it wrong. So you need to rethink what you are doing. So I think that's the the hard part of infrastructure. Scope is not a drawback, but it's trying to think simple and try to see. Think as modules and try to connect the Legos, the Lego pieces.* - J1

Code: Language issues

Another recurring theme highlighted during the interviews were the challenges associated with the HCL language that Jumia uses to define IaC. These challenges can create inefficiencies, further spreading out the limited manpower. As J3, a team lead that also takes the role of SRE states:

> *"In Jumia you have multiple ways to achieve the same goal not all of them are as efficient. So we are also working a bit on on on making things a bit more segregate and in the end more portable. And yeah, it has some somewhat of a big learning curve, but I I think that the end results are are good."* - J3

there is a somewhat big learning curve. This sentiment is also echoed by J8 *"the learning curve (is there) because the languages are very different from what the developers are used to"*. While the benefits outweigh the initial learning curve, this can still slow down adoption, and it may require additional training or input from more experienced people. This is especially evident when migrating legacy infrastructure, as previously mentioned by J1 *"that requires as well a lot of knowledge and only usually the only senior guys can do this properly without breaking anything"*.
Another challenge is the possibility for code to become outdated, as J4 mentions:

> *..."one of the drawbacks could be that sometimes this code could be a bit specific for a given version of of either AWS or some specific it was component and could become outdated and not work so well so well. Sometimes we still have the felt some limitations of of infrastructure as code as well and we had to proceed manually."* - J4

Furthermore, managing all of these dependencies can be time consuming. J4 highlights this when referring to Terraform and its plugins *"for example the specific case of Terraform we have also the not only the terraform version but also different plugins that have their own versions.) And sometimes if you want to use a new feature of AWS, you need to update also the plugin and the Terraform"*. In some cases, as J4 suggest's, it can even make manual changes the more appealing option.

Code: Resistance to change from developers

Despite the many benefits of IaC, interviews also revealed a potential challenge in overcoming the resistance to change from developers who are used to the traditional infrastructure management methods. While this may not be a universal sentiment, some developer may prefer the pre-IaC existing workflows. While J8 acknowledges this, this statement also suggest a general openness to IaC adoption.

> *"Well, you will always find people that are very keen to do things like that and people that like to be on their comfort zone. If I do TypeScript, I'm going to do TypeScript my whole life, so you will find those kind of people everywhere. But generically, there were more people willing to learn and do infrastructure as code than those that didn't ."- J8*

J1, the head of infrastructure does see an IaC benefit that helps with the adoption: *"for sure we see that the idea of them (developers) having control of some things really pleases them."*.
However, even with all the potential impact on areas like availability, monitoring and incident response, some developers have trouble getting over the drawbacks.

> *"I think it's maybe a bit hard to justify it sometimes, but it it has for sure helped for example in as we talked, things related to availability ,things to related to monitoring ,some change or if there is some incident sometimes you can check what changed or something like this".* - J4

Code: Destructive oversights

A concern presented by one of the participants was related to accidental infrastructure destruction due to errors in code. J7 warns *"Terraform is awesome but it's also requires some care because you can easily destroy all your setup with small mistakes "*.To address this issue he advocates for the importance of code review processes that can identify these errors before they are deployed.

> *"that is why I also think Terraform needs to to have peer reviews and and things like that to make sure that what is being changed is valid so we don't make the mistake of removing everything."* - J7

Chapter 7

Discussion

The discussion chapter of this thesis will delve into key findings from both the MLR and case study, bridging the gap between the established knowledge on IaC efficiency (as revealed by the MLR) and the practical aspects observed in Jumia's IaC implementation.

What we Know: Efficiency and its four outcomes. The purpose of the MLR was to define efficiency in the context of IaC, and identify key metrics used to evaluate the efficiency of IaC implementations. Through rigorous analysis, completed according to the rules presented by Garoussi et al [58] I identified four key outcomes that contribute to overall IaC efficiency:

- Speed Increase and Cost Reduction
- Scalability and Standardisation
- Security and Documentation
- Disaster Management

Furthermore, I have identified 22 key metrics across these four outcomes (as presented in 4.8), with the following five metrics being the most frequently recurring in the literature:

- Lead Time for Changes
- Time to Restore Service
- Change Failure Rate
- Deployment Frequency

What We Found: Jumia's Efficiency Perspective. The case study focused on understanding how software practitioners evaluate the efficiency of IaC implementations, and finding how the challenges and benefits associated with IaC implementations, as experienced by Jumia compare to the broader literature on the topic. Initially, the findings revealed a reactive approach to IaC, shaped by Jumia's operational priorities. While formal efficiency metrics were lacking, their focus lay on reactive measures tied to immediate infrastructure health and service availability. This reactive approach, evident in their emphasis on hardware-related metrics and rapid response procedures, underscores their primary concern – ensuring uninterrupted service, a critical aspect for an e-commerce platform.

However, when asked about the most critical metrics for evaluating IaC efficiency, several key metrics were highlighted, aligning closely with those mentioned in literature:

1. Infrastructure Code Coverage: Eessential for ensuring the stability and reliability of the infrastructure. Low coverage indicates a reliance on manual processes, hindering scalability and automation capabilities.

2. Time to Restore Service: Important for disaster recovery, a well-structured IaC foundation can significantly reduce recovery times compared to manual processes.

3. Number of Drifts: This metric, though not in the top five in literature, was emphasized by half the interviewees. It reflects the quality of IaC, with fewer drifts indicating fewer manual interventions and greater consistency in configurations.

4. Deployment Frequency: Critical for reflecting the ability to quickly adapt and implement changes. This metric indicates agility in infrastructure management.

5. Lead Time for Changes: Highlighted by half the interviewees and most mentioned in literature, this metric serves as an indicator of IaC maturity and efficiency.

6. Change Failure Rate: Emphasized for its importance in ensuring stability.Consistent environments and thorough testing are key to minimizing failure rates.

This alignment suggest a shared understanding of key IaC efficiency metrics across academia and industry, even if the initial focus may differ based on specific organization needs.
The case study also shed light on how Jumia perceives efficiency in IaC implementations. Their primary focus aligns with the traditional software engineering

definition of efficiency as presented in the MLR – achieving cost reduction and time savings. This perspective is consistent with the overall goals of IaC.
The case study also explored the challenges and benefits associated with IaC adoption at Jumia. Challenges like time investment required for migration and initial setup, as well as the initial learning curve associated with IaC adoption and specific languages highlight the complexity of transitioning to an IaC framework.
On the other hand, Jumia experienced benefits like standardisation, automation, and improved disaster recovery and reduced configuration drift, along with the IaC positive impact on fostering a DevOps culture.

Bridging the Gap. By comparing Jumia's experience with the broader literature, several key insights appear:

- While Jumia prioritizes reactive measures for immediate service needs,in line with their reactive approach to IaC, the MLR highlights the importance of a broader set of efficiency metrics. Jumia's focus serves as a reminder to tailor IaC efficiency evaluation to specific organizational goals.
- While Jumia's initial focus on non-standardized metrics didn't hinder achieving service uptime, the alignment between employee preferences and the MLR findings highlights a key point: defining IaC efficiency requires a nuanced approach. The MLR's finding's metrics shouldn't be seen as a one-size-fits-all. Instead, they should be considered alongside the specific nature and needs of the organization.

As for the benefits and drawbacks experienced by Jumia, the literature search revealed that Jumia's experience aligns with the broader understanding of IaC adoption in several ways.

Benefits:

- **Improved standardization and automation:** Similar to my findings, several studies highlight standardization (e.g., [[10], [11]] , [7]), scalability([11] , [7] , [25]),reusability of infrastructure code ([12] [25] , [57] , [10] , [36] , [20] , [7]), and enhanced transparency ([20] , [57]) as key benefits of IaC adoption. This fosters continuous improvement through infrastructure code version control ([12]) and facilitates collaboration across development teams with standardized toolsets Also mentioned was enhanced security ([36] [12])
- **Disaster recovery and reduced configuration drift:** The literature also acknowledges IaC's role in improved disaster recovery by limiting configuration drift([36]) , a common challenge in manually managed infrastructure ([36]), or in case of downed infrastructure([20] , [25] [12])

- **Fostering a DevOps culture:**The study's findings related to IaC adoption fostering a DevOps culture is echoed in literature as well, highlighting IaC's ability to foster a DevOps practices such as accountability ([7]) and collaboration([57] , [11]) , and adittionally, a reduced management overhead([10]).

Drawbacks:

- **Time investment and initial learning curve:** The literature confirms Jumia's experience regarding the time investment required for migrating legacy infrastructure [36]([36]), initial setup ([56] , [36] , [20]), and the learning curve associated with IaC adoption ([56] , [20]). Challenges related to the immutable infrastructure concept specifically were not mentioned, intead on a more broad perspective such as programming language limitations, such as constant dependency management ([[56] , [25]]), and potential resistance to change from developers have also been documented ([36]).
- **Mitigating destructive oversights:** While the findings and the interviewees emphasizes the importance of catching errors, the potential for significant consequences from mistakes in IaC configurations is a recognized drawback. Literature suggests employing rigorous testing and code review practices to minimize such risks ([12]).

However a noteworthy difference appeared regarding infrastructure monitoring. Jumia did not perceive monitoring as a significant challenge, in contrast to findings in the literature.([10] ([36]) This potential discrepancy might be attributed to Jumia's specific infrastructure or monitoring practices, a topic that could be explored in future research.

Looking Forward. While this study sheds light on IaC efficiency at Jumia, further research is needed to broaden our understanding. Future studies could explore IaC adoption across organizations with varying levels of IaC maturity. This would provide a broader picture of the challenges and benefits encountered at different stages of the IaC journey. Additionally, investigating the relationship between specific efficiency metrics and organizational performance indicators could yield valuable insights for organizations seeking to maximize the business value of IaC adoption.

Chapter 8

Threat to Validity

In this chapter, I will discuss the limitations and threats to the validity and trustworthiness of this paper. As suggested Petersen et al.[1] and Kitchenham et al.[22] , three areas of validity will be presented, and where possible, the taken actions to mitigate them.

8.1 Research design limitations

MultiVocal Literature Review The MultiVocal literature review undertaken in this paper relies over 60% on non-peer-reviewed Grey literature. Because of the high variance in writing style and information presentation, this review had to rely on a manual data extraction and analysis process that could introduce human error and bias.

Moreover, only a single author conducted the MLR, and as such there can be possible bias in the search strings, meta-data used for grey literature search on Google.com relating to browser information, location, previous search queries and machine used.

Case Study

Even though I have followed the guidelines regarding case studies, due to being a single author in this thesis, bias could be introduced. Moreso, it could be argued that the scope of a single case study is not broad enough, instead a multi-case study would fitting the research questions better.

8.2 Internal Validity

Internal validity refers to the causality of the results, or in other words, the extent to which the findings of the review can be attributed to IaC and not other factors.

MultiVocal Literature Review

Possible issues can arise from the chosen search strings, that can lead to an incomplete dataset, or from the inclusion and exclusion criteria, as they can suffer from researcher's judgement or bias. To increase internal validity, increase traceability and consistency, I strictly followed the guidelines presented by Garoussi et al. [58] in the conduction of this MLR.

Case Study

As this was a single-case study, one could argue that the scope was not broad enough to ensure internal validity. Nontheless, by correlating the findings of the literature and the case study, and finding many similarities, one could argue that the scope was broad enough. Moreso, the number of interview have allowed me to reach a satisfactory point of saturation, where the thoughts reported by the participants were consistent and in-line with each other.

8.3 Construct Validity

Construct validity refers to the extent to which the research accurately assesses the specific theoretical concept that it is intended to measure.

MultiVocal Literature Review

While the MLR followed the rigorous guidelines and standards set by Garoussi et al. throughout all of the steps, including search criteria, filtering, and data extraction, a threat to construct validity can appear from the lack of high ammount of empirical evidence used. Most of the grey literature is opinion-based based, and as such subjective. This is a limitation that cannot be fixed.

Case Study

In the case study, threats to construct validity are related to whether the selected case study actually relates to IaC, and where the interview questions were interpreted the same way by all interviewees. The threat related to the selection of the case was mitigated by following the guidelines regarding case selection [61]. As for the interviews, multiple iterations of the interview guide, tested against a developer that does not have any stake in the case study helped improve it.

8.4 External Validity

External validity refers to the extent to which the findings of the MLR can be generalized to other categories.

MultiVocal Literature Review

The search strings used in this MLR covered over 1000 artefacts, but with the limitation that all of them have to be written in English, with studies written in other languages excluded. Nonetheless, I argue that the included relevant literature does cover and contain enough information to not be required to also include other languages in this study. As for industrial context, the findings of this study are within a very specific area of study (IaC), and as such I do not intend to generalize the results to other categories.

Case Study

The findings of the case study are not meant to be generalized, but instead provide insight into the IaC implementation of Jumia. However, the results were found to be in accordance with the literature on the topic, and thus other companies implementing IaC. As such, I believe that due to these similarities, my findings would possibly be applicable to other companies in similar positions to Jumia.

8.5 Reliability

Reliability refers to the extent to which the process and findings of the thesis can be repeated.

MultiVocal Literature Review

To ensure the reliability of the study, the MLR guidelines presented by Garoussi et al. [58] were followed, with the small differences clearly presented. Moreover, all parts of the process are clearly documented in a way that is easily repeatable.

Case Study

Simmilarly, to ensure the reliability of the case study, the case study guidelines regarding case selection [61], data collection [26], triangulation [41], ethics [40], and data analysis [33] were followed.

Chapter 9

Conclusion

This thesis embarked on a two-fold journey: A multivocal Literature Review, and a Case Study.
MultiVocal Literature Review (MLR) conducted on Infrastructure as Code has two main goals: to define efficiency in the context of IaC, and identify key metrics reported in the literature to evaluate the efficiency of IaC implementations.

Following a rigorous process, as defined by Garoussi et al. [58] I identified four outcomes of efficiency in the context of IaC: "Speed Increase and Cost Reduction", "Scalability and Standardisation", "Security and Documentation", and "Disaster Management", and defined efficiency based on them. I have also identified 22 key metrics related to these outcomes, with lead time for change, time to restore service, change failure rate, deployment frequency and deployment time duration having the highest recurrence.

Furthermore, I created an artefact that presents a correlation between the metrics and the four outcomes of efficiency 4.7,that served as artefact in the second part of the thesis, a case study revolving around a company called Jumia.

The Jumia case study aimed to find how software practitioners expect to evaluate the efficiency of IaC implementations. However, at first, it revealed a perspective shaped by their operational priorities. While efficiency metrics are lacking, the emphasis lies on reactive measures related to immediate infrastructure health and service availability. This reactive approach, evident in their hardware-related metrics, and response procedures toward them underscores their focus on resolving critical issues quickly to ensure uninterrupted service, a very important goal for an e-commerce platform.
However, after interviewing the participants on their opinion on good metrics for efficiency measurement, a strong correlation between the metrics emphasized by

them, and those identified in the MLR was found. Metrics like infrastructure code coverage, time to restore service, number of drifts, deployment frequency, lead time for changes, and change failure rate were all considered critical by interviewees. This alignment suggests a shared understanding of key IaC efficiency indicators across academia and industry.
The interviewees primarily perceived efficiency in IaC implementations through cost reduction and time savings, aligning with the traditional software engineering definition of efficiency.

Lastly, the case study highlighted the commonalities between Jumia's challenges and benefits with those reported in the broader literature. Challenges like migration time, destructive oversights, initial setup complexity, and learning curve associated with IaC adoption were consistent with existing findings. Similarly, the benefits of standardization, automation, promotion of a DevOps Culture, and improved disaster recovery capabilities observed at Jumia echo the findings of literature on the topic.

Bibliography

[1] Kai Petersen et al. *Guidelines for conducting systematic mapping studies in software engineering: An update*. Sept. 18, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0950584915000646.

[2] Ricardo Amaro. *Capabilities and Practices in DevOps: A Multivocal Literature Review*. URL: https://ieeexplore.ieee.org/document/9756241.

[3] Ricardo Amaro, Rúben Pereira, and Miguel Mira da Silva. "Capabilities and metrics in DevOps: A design science study". In: *Information & Management* 60.5 (July 1, 2023), p. 103809. ISSN: 0378-7206. DOI: 10.1016/j.im.2023.103809. URL: https://www.sciencedirect.com/science/article/pii/S0378720623000575.

[4] AWS. *Metrics for everything as code*. Mar. 15, 2023. URL: https://docs.aws.amazon.com/wellarchitected/latest/devops-guidance/metrics-for-everything-as-code.html.

[5] Sebastian Baltes and Paul Ralph. *Sampling in software engineering research: a critical review and guidelines*. URL: https://doi.org/10.1007/s10664-021-10072-8.

[6] Chiradeep BasuMallick. *What Is Infrastructure as Code? Meaning, Working, and Benefits*. 2022. URL: https://www.spiceworks.com/tech/cloud/articles/what-is-infrastructure-as-code/.

[7] Chiradeep BasuMallick. *What Is Infrastructure as Code? Meaning, Working, and Benefits*. Apr. 25, 2024. URL: https://www.spiceworks.com/tech/cloud/articles/what-is-infrastructure-as-code/.

[8] Saša Baškarada. *Qualitative Case Study Guidelines*. Apr. 25, 2024. URL: https://nsuworks.nova.edu/cgi/viewcontent.cgi?article=1008&context=tqr.

[9] Garth Booth. *Box CMF: DevOps DORA, Infrastructure as Code*. URL: https://medium.com/box-tech-blog/box-cmf-devops-dora-infrastructure-as-code-9459ecb2161b.

[10] Cloudbolt. *3 Advantages and Challenges of Infrastructure as Code (IaC)*. Apr. 25, 2024. URL: https://www.cloudbolt.io/blog/3-advantages-and-challenges-of-infrastructure-as-code-iac/.

[11] CodeFresh. *Infrastructure as Code: Benefits, Platforms Tips for Success*. Apr. 25, 2024. URL: https://codefresh.io/learn/infrastructure-as-code/.

[12] DuploCloud. *The 7 Biggest Benefits of Infrastructure as Code*. Apr. 25, 2024. URL: https://duplocloud.com/blog/infrastructure-as-code-benefits/.

[13] Justin Ellingwood. *Gathering Metrics from Your Infrastructure and Applications*. Dec. 26, 2017. URL: https://www.digitalocean.com/community/tutorials/gathering-metrics-from-your-infrastructure-and-applications.

[14] Breno Nicolau de França. *Characterizing DevOps by Hearing Multiple Voices*. URL: https://www.researchgate.net/publication/308855773_Characterizing_DevOps_by_Hearing_Multiple_Voices.

[15] D. Giustini. *Finding the Hard to Finds: Searching for Grey (Gray) Literature UBC*. URL: hlwiki.slais.ubc.ca/images/5/5b/Greylit_manual_2012.doc.

[16] H. Huijgens et al. "Factors Affecting Cloud Infra-Service Development Lead Times: A Case Study at ING". In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). May 25, 2019, pp. 233–242. DOI: 10.1109/ICSE-SEIP.2019.00033.

[17] Alistair Heys. *Use Infrastructure as Code (IaC) to make your team and customers happy*. Mar. 1, 2023. URL: https://www.scalr.com/blog/use-iac-to-create-happiness.

[18] *Infrastructure-as-code*. 2022. URL: https://www.rapid7.com/fundamentals/what-is-infrastructure-as-code-iac/.

[19] Jumia. *Jumia technologies is the best of amazon in africa*. URL: https://group.jumia.com/news/jumia-technologies-is-the-best-of-amazon-in-africa.

[20] Kihara Kimachia. *Benefits and Drawbacks of Infrastructure as Code (IaC)*. Apr. 25, 2024. URL: https://www.enterprisenetworkingplanet.com/data-center/infrastructure-as-code/.

[21] B. Kitchenham and S. Charters. *Guidelines for Performing Systematic Literature Reviews in Software engineering*. URL: EBSETechnicalReport2007volEBSEâĂŘ2007âĂŘ01.

[22] Barbara Kitchenham and Stuart Charters. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Sept. 18, 2023. URL: https://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf.

[23] John Klein. *INFRASTRUCTURE AS CODE–FINAL REPORT*. URL: https://insights.sei.cmu.edu/documents/576/2019_019_001_539335.pdf.

[24] Dmitriy Konstantynov. *INFRASTRUCTURE AS CODE IN DEVOPS*. Aug. 10, 2020. URL: https://alpacked.io/blog/infrastructure-as-code-for-devops/.

[25] Kedar Vijay Kulkarni. *The pros and cons of infrastructure-as-code*. Apr. 25, 2024. URL: https://www.redhat.com/sysadmin/pros-and-cons-infrastructure-code.

[26] Timothy Lethbridge, Susan Sim, and Janice Singer. *Studying Software Engineers: Data Collection Techniques for Software Field Studies*. Sept. 18, 2023. URL: https://www.researchgate.net/publication/220277828_Studying_Software_Engineers_Data_Collection_Techniques_for_Software_Field_Studies.

[27] Martin Forsberg Lie, Mary Sánchez-Gordón, and Ricardo Colomo-Palacios. "DevOps in an ISO 13485 Regulated Environment: A Multivocal Literature Review". In: *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ESEM '20. New York, NY, USA: Association for Computing Machinery, 2020. ISBN: 978-1-4503-7580-1. DOI: 10.1145/3382494.3410679. URL: https://doi.org/10.1145/3382494.3410679.

[28] linkedin. *Jumia*. Apr. 25, 2024. URL: https://www.linkedin.com/company/porto-tech-center/.

[29] DAVID LJUNGGREN. *DevOps: Assessing the Factors Influencing the Adoption of Infrastructure as Code, and the Selection of Infrastructure as Code Tools*. Nov. 3, 2021. URL: https://kth.diva-portal.org/smash/get/diva2:1792965/FULLTEXT01.pdf.

[30] Lucy Ellen Lwakatare, Pasi Kuvaja, and Markku Oivo. *Relationship of DevOps to Agile, Lean and Continuous Deployment*. Sept. 18, 2023. URL: https://www.researchgate.net/publication/309711040_Relationship_of_DevOps_to_Agile_Lean_and_Continuous_Deployment.

[31] Lucy Ellen Lwakatare et al. "DevOps in practice: A multiple case study of five companies". In: *Information and Software Technology* 114 (Oct. 1, 2019), pp. 217–230. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2019.06.010. URL: https://www.sciencedirect.com/science/article/pii/S0950584917302793.

[32] Mudit Mathur. *Understanding Infrastructure as Code and Configuration Management*. June 30, 2023. URL: https://muditmathur121.medium.com/understanding-infrastructure-as-code-and-configuration-management-558f6d0659d5.

[33] Matthew Miles, Michael Huberman, and Johnny Saldana. *Qualitative Data Analysis A Methods Sourcebook*. URL: https://us.sagepub.com/en-us/nam/qualitative-data-analysis/book246128.

[34] Kief Morris. *Infrastructure as code*. Nov. 15, 2020. URL: https://www.oreilly.com/library/view/infrastructure-as-code/9781098114664/ch01.html.

[35] Kief Morris. *Infrastructure as code*. Nov. 15, 2020. URL: https://www.oreilly.com/library/view/infrastructure-as-code/9781098114664/ch01.html.

[36] Olga Murphy. *Adoption of IaC in Real World*. Apr. 25, 2024. URL: https://www.theseus.fi/bitstream/handle/10024/786729/Thesis_Murphy_Olga_YTS20K1.pdf?sequence=2.

[37] R. T. Ogawa and B. Malen. *Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method*. URL: https://www.jstor.org/stable/1170630.

[38] Akond Rahman, Rezvan Mahdavi-hezaveh, and Laurie Williams. *A Systematic Mapping Study of Infrastructure as Code Research*. URL: https://www.researchgate.net/publication/329598845_A_Systematic_Mapping_Study_of_Infrastructure_as_Code_Research.

[39] reintech.io. *Efficiency*. Apr. 25, 2024. URL: https://reintech.io/terms/category/efficiency-software-development.

[40] JANe RITCHIe and JANE LEWIS. *QUALITATIVE RESEARCH PRACTICE*. URL: https://mthoyibi.wordpress.com/wp-content/uploads/2011/10/qualitative-research-practice_a-guide-for-social-science-students-and-researchers_jane-ritchie-and-jane-lewis-eds_20031.pdf.

[41] Höst M Runeson P. *Guidelines for conducting and reporting case study research in software engineering. Empir Software Eng 14, 131–164 (2009)*. URL: https://doi.org/10.1007/s10664-008-9102-8.

[42] Per Runeson et al. *Case Study Research in Software Engineering: Guidelines and Examples*. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118181034.

[43] S. Muthoni, G. Okeyo, and G. Chemwa. "Infrastructure as Code for Business Continuity in Institutions of Higher Learning". In: *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET). Dec. 9, 2021, pp. 1–6. DOI: 10.1109/ICECET52533.2021.9698544.

[44] Johnny Saldana. *The coding manual for qualitative researchers*. URL: https://us.sagepub.com/.

[45] Carlos Schults. *What Is Infrastructure as Code? How It Works, Best Practices, Tutorials*. Sept. 18, 2023. URL: https://stackify.com/what-is-infrastructure-as-code-how-it-works-best-practices-tutorials/.

[46] Mali Senapathi, Jim Buchan, and Hady Osman. "DevOps Capabilities, Practices, and Challenges: Insights from a Case Study". In: *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. EASE '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 57–67. ISBN: 978-1-4503-6403-4. DOI: 10.1145/3210459.3210465. URL: https://doi.org/10.1145/3210459.3210465.

[47] Adarsh Shah. *Infrastructure as Code: Principles, Patterns, and Practices*. URL: https://www.cloudknit.io/blog/principles-patterns-and-practices-for-effective-infrastructure-as-code.

[48] Musarrat Shaheen, Pradhan Sudeepta, and Ranajee Ranajee. *Sampling in Qualitative Research*. URL: https://www.researchgate.net/publication/345131737_Sampling_in_Qualitative_Research.

[49] Dustin Smith. *State of DevOps 2021*. Nov. 3, 2021. URL: https://services.google.com/fh/files/misc/state-of-devops-2021.pdf.

[50] Dustin Smith. *State of DevOps 2022*. Nov. 3, 2021. URL: https://services.google.com/fh/files/misc/2022_state_of_devops_report.pdf.

[51] Sonar. *infrastructure as code (IaC): basic beginner's guide*. 2023. URL: https://www.sonarsource.com/learn/infrastructure-as-code/.

[52] stackoverflow. *What's the average tenure of an engineer at a big tech company?* Apr. 25, 2024. URL: https://stackoverflow.blog/2022/04/19/whats-the-average-tenure-of-an-engineer-at-a-big-tech-company-ep-434/.

[53] SYSE. *IT efficiency*. 2023. URL: https://www.suse.com/suse-defines/definition/it-efficiency/.

[54] Mia-Platform Team. *DevOps Metrics: how to monitor performances optimally*. Nov. 3, 2021. URL: https://mia-platform.eu/blog/devops-metrics/.

[55] E. Tom, A. Aurum, and R. Vidgen. *An exploration of technical debt*. URL: JournalofSystemsandSo 86,pp.1498-1516,2013..

[56] Christopher Tozzi. *Infrastructure as Code: Do the Drawbacks Outweigh the Benefits?* Apr. 25, 2024. URL: https://www.itprotoday.com/software-development/infrastructure-code-do-drawbacks-outweigh-benefits/.

[57] Jason Turim. *The Pros and Cons of IaC: What You Need to Know*. Apr. 25, 2024. URL: https://thenewstack.io/the-pros-and-cons-of-iac-what-you-need-to-know/.

[58] Garousi V., Felderer M., and Mäntylä. *TGuidelines for including grey literature and conducting multivocal literature reviews in software engineering*. URL: https://pureadmin.qub.ac.uk/ws/portalfiles/portal/178657399/GuidelinesforconductingMLRs_Sept18.pdf.

[59] Stephen Watts. *2023 State of DevOps*. URL: https://www.splunk.com/en_us/blog/learn/state-of-devops.html.

[60] Robert K. Yin. *Applications of Case Study Research*. URL: https://us.sagepub.com/en-us/nam/node/54068/print.

[61] Robert K. Yin. *Case Study Research Design and Methods*. URL: https://www.sagepub.com/sites/default/files/upm-binaries/24736_Chapter2.pdf.