Digital security monitoring of cloud services

SLA Analyser- Availability Monitoring tool for DB Cloud Services

Mosammath Nazifa Anjum Islam Anika
(nazifaanjumanika@gmail.com)
Sneha Gupta
(eu.sneha@gmail.com)

Department of Computer
and Systems Sciences

Degree project 30 HE credits
Information Security
Degree project at the master level
Autumn term 2023
Supervisor: Elias Seid (elias.seid@dsv.su.se)

Stockholm
University

Abstract

Cloud computing is in current demand and widely used for businesses because of its scalability and flexibility. An agreement is signed between cloud providers and customers to guarantee the level of services are available as promised, it is called Service Level Agreement (SLA). To assess if the services are compliant with SLA, availability of the cloud services is important to be monitored. There are some existing monitoring tools provided by the cloud service providers. However, those can be often biased that leads to calls for independent SLA evaluation. This study aims to solve the problem of verifying service availability parameters in terms of uptime and downtime against SLA by proposing a monitoring tool- SLA Analyser. This paper focuses on reliable assessment of SLA for both customers and service providers. Furthermore, this research discusses the challenges of SLA tracking and consequences of not assessing it. This approach is even more essential for organizations that use multiple cloud services from different vendors. The SLA Analyser was utilised to monitor the availability of the cloud database services. To avoid business losses and maintain reputation for customer trust, it is crucial to validate the SLA. The research problem related to SLA monitoring is challenging in terms of relevant previous studies of this area. Using experiment-based research strategy and quantitative data analysis method, the results have shown that the proposed solution is able to independently monitor the service availability and validate with SLA for cloud databases. The recommended solution is of significant interest to organizations as they are migrating and using single or multiple cloud services by different vendors as per the business need. The SLA Analyser serves as a valuable tool for both customers and service providers to solve issues related to availability monitoring of the cloud database services and ensure the SLA commitment is met.

Keywords: Availability, Cloud database services, Monitoring, SLAs

Synopsis

Background	The current situation is that SLA documents are complex and ambiguous. It may define terms such as availability and security in specific and limited ways. With cloud providers delegating the task of SLA violation to the consumers, it becomes difficult as cloud consumers hardly detect SLA violations since detection of breach is a subjective and complex task. Often customers get to know about the downtime of their cloud hosted application when the users complain about accessibility. Many cloud providers offer monitoring tools to enable customers to evaluate the performance of their cloud services. However, these tools are provider specific and cannot be customized to meet client's specific requirements. In addition, such monitoring tools are provider-dependent and the same monitoring platform cannot be used for multiple cloud providers.
Problem	Several activities have been carried out for the defining SLA models and metrics. There are also many tools available in the market for cloud monitoring. These cloud monitoring tools help in managing, monitoring the performance of the cloud infrastructure, services and applications. As each tool has its pros and cons, but they all aim to help customers gain observability and monitor performance of the cloud services. Most importantly, being aware of the system's current software and hardware service status is crucial for ensuring the fulfillment of targets as per the SLA document (Alhamazani et al., 2014). Currently, SLA-based Cloud security monitoring services and tools are not yet available (Dana et al., 2014). In order to address these shortcomings, we present this research paper.
Research Question	The research focuses on utilising SLA Analyser, a monitoring tool to independently monitor the availability parameter of SLA for cloud based database-as-a-service.
Method	For this paper, experiment-based research is chosen as the research strategy. Researchers experiment with a prototype that helps to create a controlled situation to investigate the availability of cloud database services and to discover if the availability matches with the percentage agreed upon in the Service Level Agreement. The experiment-based data collection method is selected as it aligns with the nature of aim and research questions. The data generated during this experiment by the tool proposed is further utilised to generate the SLA compliance report. A quantitative data analysis method has been chosen as data analysis method.
Result	While conducting the experiment, four main experimental scenarios were created: *Scenario1*: SLA Analyser tool shall monitor the available database cloud service *Scenario2*: SLA Analyser tool shall monitor the unavailable database cloud service *Scenario3*: SLA Analyser tool shall monitor more than one database cloud services

	Scenario4: SLA Analyser tool shall generate SLA compliance reports for database cloud services The tool fulfilled and passed all the scenarios and was able to measure the availability parameters of an SLA for cloud based database-as-a-services.
Discussion	The experiment and analysis provide sufficient representation for answering the research question of this paper. The SLA Analyser that is programmed using C# is utilised to independently monitor the availability parameter of SLA for cloud based database-as-a-service. The availability and unavailability of the database service was collected and compared with the SLA availability percentage. The analysis shows how the tool could capture the unavailability of the cloud services against the promised percentage of SLA for some days and also how the tool shows that the services have been available in terms of the Service Level Agreement on the other days.

Acknowledgement

We want to extend our sincere gratitude to all those who have supported us while completing this master's thesis. First and foremost, we sincerely appreciate our supervisor, Elias Seid for his valuable guidance, expertise, knowledge and continuous support throughout this research. Elias’s input and feedback have significantly contributed to the quality and development of this thesis. We appreciate the helpful feedback from the Reviewer, Jakob Tholander and, also would like to thank the Examiner for taking time to read and grade the paper.

We are also grateful to the faculty and staff of DSV, Stockholm University for providing a conducive learning environment and valuable courses that have helped us to pursue our research effectively. Our thanks go to all the participants of this study, who shared their insights and feedback through peer review and seminars that has greatly improved our research work.

Finally, we thank all those who have provided assistance, whether through discussions, proofreading, or technical support. All the contributions have played an essential role in shaping this thesis. We are sincerely thankful to everyone who has been part of our journey, believing in our abilities and supporting us along the way.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AZs: Availability Zones
CIA: Confidentiality, Integrity, Availability
CSPs: Cloud Service Providers
DB: Database
DBaaS: Database as a Service
DBMS: Database Management System
HA: High Available
Iaas: Infrastructure as a Service
IWGCR: International Working Group on Cloud Computing Resiliency
ITC: Information and communication technologies
KPIs: Key Performance Indicators
PaaS: Platform as a Service
QoS: Quality of Services
REST: Representational State Transfer
SaaS: Software as a Service
SLA: Service Level Agreement
SLOs: Service Level Objectives

1 Introduction

In recent years, Cloud Service has become a popular topic in the Information and communication technologies (ITC) domain. The term "cloud services" refers to a wide range of services delivered on demand basis over the internet by the vendors to companies and customers. Many companies and institutions are moving in various ways to leverage these services. These cloud services are designed to provide its customers, an easy, scalable, cost effective access to resources, without being involved in the maintenance of internal infrastructure or hardware.

Cloud services are managed by cloud computing vendors and service providers. They are made available to customers from the providers' servers. Service owners and operators need to keep track of the running status of each component of the cloud infrastructure for faults, usage, and to ensure that business operations stay uninterrupted for use. Analytical information in a cloud-based service infrastructure is achieved on the basis of logs from the running components of the designed system. For example, utilizing logs, one can determine the availability of the service infrastructures hosted on cloud. This information about availability of components can help organizations in making calculative and risk based choices between cloud components and vendors.

A cloud service level agreement (also called SLA) ensures cloud providers meet certain enterprise-level requirements and provide customers with a clearly defined set of deliverables. SLAs may include measuring Quality of Service (QoS), which are agreed between the vendor and an organization (customer). The availability of the cloud infrastructure is also covered under the SLA, where the service provider and their end-user defines the expected level of cloud-based service infrastructure uptime and accessibility. The vendor is to ensure and outline the penalties involved, usually financial if the agreed-upon service levels are not met.

The current situation is that SLA documents are complex and ambiguous. It may define terms such as availability and security in specific and limited ways. Additionally, service agreements often place differing responsibilities on consumers to track changes in service agreements and to determine when to re-evaluate service agreements (Badger et al., 2012). As Barros said in his book, with more external service vendors involvement, the lesser control a customer has over the quality of the delivery of these services. Thereby the customer must rely on the quality commitments agreed as per the SLAs.

The major drawback is that the cloud computing providers delegate the task of providing a service violation to the cloud consumers (Barros, 2015). Cloud consumers hardly detect SLA violations since detection of breach is a subjective and complex task. Often customers get to know about the downtime of their cloud hosted application when the users complain about accessibility. The unavailability of the service noticed by its users brings bad reputation to the customer's business. Many cloud providers offer monitoring tools to enable customers to evaluate the availability of their cloud services. However, these tools are provider specific and cannot be customized to meet client's specific requirements. In addition, such monitoring tools are provider-dependent and the same monitoring platform cannot be used for multiple cloud providers.

Several activities have been carried out recently, in the context of academic research and industry-driven initiatives on the defining SLA models and SLA performance metrics. There are also many tools available in the market for cloud monitoring. These cloud monitoring tools help in managing, monitoring the performance of the cloud infrastructure, services and applications. Some of the available cloud monitoring tools in the market are Monitis, RevealCloud, LogicMonitor, etc (Alhamazani et al., 2014; Kuć, 2023).

Each tool has its pros and cons, but they all aim to help customers gain observability and monitor performance of the cloud services. Most importantly, being aware of the system's current software and hardware service status is crucial for ensuring the fulfillment of targets as per the SLA document (Alhamazani et al., 2014). Currently, SLA-based Cloud security monitoring services and tools are not yet available (Dana et al., 2014). In this research, SLA Analyser tool is designed to be utilized to monitor the availability of cloud database service.

A cloud database is a database that typically runs on a cloud computing platform and access to the database is provided as-a-service. This cloud computing service is referred to as Database-as-a-service (DBaaS), also known as managed database service. DBaaS enables customers to access and use a cloud database system without having to buy and set up their own hardware, install their own database software, or manage the database themselves. Periodic upgrades, backups, and ensuring that the database system is available and safe are all handled by the cloud provider.

A wide range of cloud database services are sold in the market which are highly available and production ready. Though databases are delivered and consumed independently, their pricing and interaction models can be different. DBaaS offerings provide enterprise grade security, flexibility, scalability, and availability based on customer requirements. These requirements are agreed upon and signed under SLA documents by the providers and the customer. SLA Analyser aims to help cloud consumers monitor the availability of their cloud database and detect violations of SLA documents.

1.1 Research Problem

When it comes to cloud computing, the availability of a service is critical to ensure that customers can access the cloud services whenever they need to and from any location around the globe. To ensure availability, the cloud service providers guarantee that the services will be available and there will be very little downtime for the services. Service level agreements are used to formally establish the service parameters like availability, quality, and more, that a provider commits to delivering. The monitoring of these service parameters constantly is crucial. Depending on the SLA definitions, the customer may also be qualified to get some payment (in the form of penalty) if an SLA violation is discovered. Thereby reliable validation of SLA agreements is of paramount importance for both cloud service providers and consumers.

The fact that most of the service parameters (and hence SLAs) are measured by the monitoring tools which are provided by service providers themselves establishes chances for a bias. For such an important control of compliance, it is important to have assurance via an independent review. The consumers today have no other option than to trust their suppliers' statistical reports showing their compliance to the promised SLAs. This is the reason why a trustable and independent monitoring of SLA is needed. The research aims to propose a solution to this problem.

Besides, the customers who are paying for these cloud services are running their businesses by hosting their applications, services, etc on cloud. For these customers it becomes extremely important to ensure that these cloud services are up and running at all times. Any failure to this can damage their reputation in the market and can lead to loss of business. Therefore the availability monitoring of the cloud services and validating against SLA is significantly important for both the cloud providers and the customers.

As stated by Faction Inc. & Amanda, 94% of companies use cloud services in 2022. This massive percentage is partially due to a shift that occurred due to the COVID-19. The pandemic led 61% of businesses to migrate their workloads to the cloud. Further, it is estimated that cloud usage will continue to rise even more than the current rate of 94% (Faction Inc. & Amanda, 2022). Due to the dynamic cloud character, the service attributes must be monitored and managed consistently (Frey et al., 2013). As companies are migrating and are using multiple cloud services by multiple vendors based on their business needs, solving the issues related to availability of these cloud services and their respective SLA monitoring is of general and significant interest.

In order to be able to verify service parameters against SLAs, SLA in the first place needs to be simple and written in a quantified way that is easily understood (Frey et al., 2013). The research problem related to SLA tracking is also challenging as it is currently very time-consuming and requires pulling lots of quality metrics and validating them against agreed cloud services SLA. This becomes even more demanding for organisations which are using multiple cloud services from different vendors.

Therefore as stated above, the research problem is general, significant and challenging. Though it is not unique, being able to monitor the availability of a cloud service holds great importance in the ITC domain. Utilizing a tool like SLA Analyser will help both the cloud provider and the customers to independently track the availability of their cloud services against the SLA documents.

1.2 Aim and Research Question

The aim of the research is to utilize a proposed tool to monitor the availability aspect in the SLA of public cloud infrastructure components. The proposed tool is named as SLA Analyzer which monitors the availability of cloud service. This tool has two parts, a core application which generates the data and collects them in the main registry database and the second part is the compliance report generator, which calculates the availability of the cloud services based on the data collected previously and compares the availability against the SLA availability parameter. Based on the data collected and analyzed, a graphical form of report is generated.

Downtime of a service is the time during which a service is unavailable for use and the uptime of a service is the time during which a service is available for use. The SLA Analyzer generates information about the downtime and uptime of the cloud service. This data will also be utilized for statistical analysis to calculate the net availability of the cloud service. Availability is calculated based on the downtime and uptime ratio as below (Oliveto, 1999):

$$Availability = uptime/(downtime + uptime)$$

The calculated *availability* is then validated against the values of availability parameters, as mentioned in the SLA documents which was agreed between the cloud service providers and the customer. The research will propose SLA Analyzer tool and utilize it to independently monitor the availability parameter of SLA for cloud service.

Practically there are several cloud services to make a choice from. And each of these services have their own SLA based on the availability zone. This means the research can be extended for measuring SLA for multiple cloud services. For the scope of this research, SLA of database cloud services will be assessed. There are few reasons behind this selection. Firstly, the usage of the database cloud services is very common. Most of the organizations manage data by using different kinds of database cloud solutions and hence it is a popular choice. Secondly, there are many supporting research work related to service parameters of database cloud services (Manuel, 2013; Sousa et al., 2012) which will be useful while implementing the monitoring tool.

This research will focus on measuring the availability parameter of SLA for cloud services related to databases. It uses the data generated by the SLA Analyser tool to do the analysis and calculations for the availability of the database services. Thereby the research tries to establish answers to the below research question.

RQ:

How to utilize SLA Analyzer tool to independently monitor the availability parameter of SLA for cloud based database-as-a-service?

1.3 Delimitations of the Study

By design, this study holds particular delimitations. Firstly, among all the different cloud services present in the market, this research centers on the database cloud services in order to be able to have focused investigation in one chosen area.

Secondly, the results of the research are based on the data sample collected through experiments on a prototype cloud database service used by the researchers associated with the study, not from any practical organization. Though practical organizations use the service in larger terms, the availability monitoring concepts should still hold applicable. Hence, the results with the collected data are expected to be applicable for different organizations and for different cloud services.

Lastly, the proposed tool development, experiment and validation have been done by authors of this research. This could hold some biases as there was no external group for validating the experiment. This choice was made, taking into consideration the limited resources and time available for conducting the research. Time and resources constraints did not permit a longitudinal study, which would include other external sources or people. Though, measures were taken to validate the results through statistical analysis and acceptance scenarios to come to a fair conclusion to answer the research question.

2 Background

The term "service-level agreement" (SLA) refers to a legal agreement between a service provider and its clients that outlines the services they will deliver and the service standards they are required to uphold. Service providers require SLAs to manage customer expectations, specify the severity levels, and spell out the conditions in which they are not responsible for outages or poor performance. Customers can also gain from SLAs because the agreement outlines the service's performance parameters, which can be compared to those of competing suppliers, and specifies how service problems will be resolved (Rosencrance et al., 2021).

As managed services and cloud computing services become more prevalent, SLAs evolve to address the new approaches. The negotiation of computation and performance requirements between a service provider and user typically takes place through the use of SLAs. Key components of a service-level agreement include- Agreement overview, Description of services, Exclusions, Service performance, Compensation, Stakeholders involved, Security, Risk management and disaster recovery, Service tracking and reporting, Periodic review and change processes, Termination process and Signatures. To enforce a service-level agreement, it is important to understand the provider's service delivery standards. The client may be entitled to the compensation specified in the contract if the SLA is not being effectively met. In order to measure the service provider's performance, SLAs include certain metrics and parameters. Some of these parameters are- Availability and Uptime Parameters, Performance Parameters, Data, Security and Privacy, Hardware and Software Requirements, etc.

The availability and uptime parameters is the amount of time services are running and accessible to the customer. For cloud services, uptime and availability are very important parameters. This is particularly relevant given that an application's availability may depend on the functionality of the underlying infrastructure; thus, an application may be running but yet be unavailable if there is a problem with the infrastructure that prevents it from being provided. A digital application cannot be used if it is not accessible and available. These parameters are critically defined based on the type of cloud service- Software as a Service (SaaS), Infrastructure as a Service (Iaas), or Platform as a Service (PaaS) products.

According to a study from 2019 named ‘Cloud security service level agreement: Representation and Measurement’, there are no standardized systems in place for cloud computing to define and enforce security SLAs (Hubballi et al., 2019). Cloud service providers (CSPs) must provide guarantees for the security and computation used for critical applications. Typically, customers negotiate Service Level Agreements to enforce these obligations. Platforms for cloud computing now use different tools or mechanisms given by service providers for measuring and enforcing SLAs related to performance. However, these SLAs typically do not address security-related issues and challenges (Hubballi et al., 2019). Instead, CSPs employ security and openness standards like the Cloud Control Matrix (CCM) from the Cloud Security Alliance and the NIST's SP 800-53 (Hubballi et al., 2019). The availability of the service, privacy of data, transparency, billing, and preventing cyberattacks are some of the security concerns with cloud computing. Due to the heterogeneous nature of the cloud, which is made up of several types of computer systems, maintaining the security of cloud services to cover all the aspects of security is challenging (Hubballi et al., 2019).

SLAs are frequently used to define and oversee the QoS (Quality of Service) agreements made between clients and service providers. According to (Kaaniche et al., 2017), there aren't many SLA management systems that take cloud security environments into account. Additionally, in many systems, operational performance management is more advanced than security management (Kaaniche et al., 2017). In a SLA, security duties related to a service should be stated properly, for instance confidentiality, integrity and availability (CIA). Furthermore, the study proposed to take into account the dynamic nature of cloud systems during formulation of security SLAs for proper monitoring (Kaaniche et al., 2017).

In a study from 2017 stated it is difficult for providers to provide clients with trustworthy services due to the lack of security management in the present SLAs and the lack of tools for making unbiased comparisons between various service offers (Kaaniche et al., 2017). According to the knowledge of the article, there are hardly any proper tools yet to provide a comprehensive mechanism for describing security SLAs completely and managing them in real-time (Kaaniche et al., 2017).

A study on the adoption of security SLAs in the cloud explained that cloud services providers (CSPs) usually offer security that is not very transparent (Casola et al., 2015). This means that the security mechanisms are vulnerable and not very flexible to negotiation. This means that customers have few choices when it comes to security features (Casola et al., 2015). However, it would be desirable for security to be treated just as other important factors when choosing a cloud service. In fact, the European Community has launched a number of projects to establish a semantic and common understanding of SLAs for cloud computing (Casola et al., 2015). According to the study by (Casola et al., 2015), specific subgroups are working on security-related issues, but as of right now, security is not up to date.

Recently, more and more businesses have started looking into how to take advantage of the pay-per-use model and quick elasticity offered by cloud computing(Undheim et al., 2011). But the significant obstacles related to security must be overcome so that businesses can trust cloud providers with their essential business applications. According to a study, these difficulties mostly concern QoS, including dependability, availability, performance, and security (Undheim et al., 2011). The current (SLA) is insufficient to address all of these issues properly (Undheim et al., 2011).

A model for remotely accessible "as-a-service" provision of resources (storage, apps, etc.) is given by cloud computing (Taha et al., 2020). The frequency of security breaches is still increasing despite the many promised advantages of the Cloud for maintaining the confidentiality, integrity, and availability of the stored data (Taha et al., 2020). This study also stated that customers and businesses were unable to trust the cloud service providers completely due to a lack of security assurance and transparency (CSPs). Customers cannot be certain that the CSPs will meet their criteria unless the CSPs identify and document the customer's security requirements accurately (Taha et al., 2020) . Additionally, compensating the customer after a violation requires a manual, time-consuming process for this reason (Taha et al., 2020).

Additionally, the SLA contains a set of Service Level Objectives (SLOs), which are the quantifiable components of a SLA that outline the levels of cloud services expected by customers and necessary for the CSP to meet. According to the study, despite having the direct economic value, taking full advantage of cloud computing requires a significant acceptance of off-the-shelf services (Taha et al., 2020). In particular, security assurance and transparency continue to be two of the key conditions for enabling customers' faith in CSPs (Taha et al., 2020). A study from 2020 states that cloud clients frequently struggle to evaluate the security of the CSP(s) they are paying for due to a lack of assurance and transparency as well as the existing scarcity of methods to quantify security (Taha et al., 2020). In a study related to decentralized runtime monitoring approach relying on the Ethereum Blockchain Infrastructure explained that a number of cloud community stakeholders are pushing for the inclusion of security parameters and CSP's security implementation in service level agreements in this context that named secSLA (Taha et al., 2020). But it is not widely used yet. Also, there is still room for improvement for monitoring according to the study (Taha et al., 2020).

Like several different services, cloud provides database management facilities. Fundamentally a database is an organized collection of data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Rather than being installed and maintained on-premises by an organization, a database in cloud services refers to a DBMS that is hosted and run by a cloud service provider. A cloud database service is nothing more than a database service that is accessible and available from anywhere.

Since cloud databases are deployed in a cloud environment, no dedicated hardware is required to host a database while using a cloud database. The cloud provider can provide, operate, and scale the underlying database cluster instead of the customers installing, configuring, and maintaining database instances themselves. The database itself can either be hosted in a cloud-based virtual machine or made available as a SaaS application. Applications can then access the database over a network from any device.

Cloud databases host and share information between several devices mostly through internet communication, and it is anticipated that the number of these devices will grow (Januzaj et al., 2015). Many businesses now provide these cloud database services including Microsoft Azure, Google, Amazon EC2, GoGrid, Garantia Data, Mongo Lab, and others (Januzaj et al., 2015). With the help of these services, businesses can quickly install, maintain, and grow their databases without worrying about the underlying infrastructure(Capel et al., 2020).

These businesses provide cloud services with the two common deployment models: we may use a database without a virtual machine or we can buy a cloud service database that is maintained by the cloud companies listed. The secondary model is referred to as Database-as-a-service (DBaaS). Because they offer services under the "pay as you go" model, which makes services cheaper than other services, these cloud services are considerably more suitable for end customers (Januzaj et al., 2015).

Compared to deploying a database management system on-premises, DBaaS offers your organization significant benefits like- including the minimization of infrastructure resources costs, and its elasticity property, which allows services to be scaled up or down according to the current demand, etc. Despite these perks, cloud databases have some challenges. Those are mainly management and security issues (Januzaj et al., 2015). From the Cloud provider point-of-view, there are many challenges to be overcome in order to deliver Cloud services that meet all requirements defined in Service Level Agreements (SLAs). High availability has been one of the biggest challenges for providers, and services like- checkpointing, load balancing, and redundancy can be used to improve the availability of a service (Endo et al., 2016).

A report from the International Working Group on Cloud Computing Resiliency (IWGCR) gathers information regarding services downtime and associated revenue losses. It points out that Cloud Foundry downtime results in $336,000 less revenue per hour. Paypal, the online payment system, experiences a revenue loss of $225,000 per hour. To mitigate the outages, Cloud providers have been focusing on ways to enhance their infrastructure and management strategies to achieve high available (HA) services (Cérin et al., 2013).

According to Toeroe & Tam availability is calculated as the percentage of time an application and its services are available, given a specific time interval. One achieves high availability (HA) when the service in question is unavailable less than 5.25 minutes per year, meaning at least 99.999 % availability (2012). Delivering a higher level of availability has been one of the biggest challenges for cloud providers. In order to reach a certain level of high availability, a cloud provider should monitor its resources and deployed services continuously (Endo et al., 2016).

Availability is one of the core security parameters of CIA(Confidentiality, Integrity, Availability) which means the system should be available during the agreed-upon time to the designated people (Januzaj et al., 2015). In this study, we want to monitor availability of DBaaS. Our tool checks with a neutral perspective if the agreed-upon availability of the Database cloud services provided by specific/multiple cloud providers is met and notify if SLA is violated at any point in the service lifecycle.

3 Research Methodology

This chapter presents the method choice for research strategy, data collection, and data analysis for answering the research question. It explains the reasons behind the selection of these methods by discussing possible methods and their advantages and disadvantages given the research question before concluding what the best options are. The selection of the informants and the text about the ethical issues considered regarding the research are also included in this chapter.

3.1 Research Strategy

Denscombe (2010) stated that the right research strategy which is fit to the purpose is important to choose in relation to the particular objective the research is trying to achieve. There are advantages and disadvantages to each choice. However, it is important that the choice of strategy can be justified in terms of being feasible, being ethical, and as providing suitable kinds of data for answering the research question (p. 4). For this paper, experiment-based research is chosen as the research strategy as the purpose of this strategy is discovering new properties/relationships associated with the materials being investigated or testing existing theories (Denscombe, 2010).

Comparison between different strategies shows that experiment-based research approach is the most suited method considering the nature of this study. Denscombe (2010) defines experiment-based research approach as an empirical examination conducted under controlled conditions with the goal of determining the characteristics of or establishing a link between certain components. In our study, the researchers experiment with a prototype that helps to create a controlled situation to investigate the availability of cloud service databases and check with the SLA to discover if the availability matches with the percentage agreed upon in the Service Level Agreement. Three main things have to be followed in terms of conducting an experiment as stated in Denscombe (2010). Those are identification of causal factors, controls and empirical observation and measurement.

Identification of factors enables the researchers to pinpoint which factors cause the observed outcome to occur (Denscombe, 2010). In our study, we identified the availability factor of cloud services through which we can figure out if the service is provided in terms of SLA. By controls, it means that researchers need to identify the significant factors and then either introduce or exclude those to observe the effect properly (Denscombe, 2010). Availability is chosen to cross-check with the SLA and other factors were excluded to discover clearly if the service availability matches with the agreed upon percentage. Empirical observation and measurement is an essential step of experiment-based research strategy (Denscombe, 2010).

Research strategies, such as grounded theory, case study and phenomenology are rejected because these strategies demonstrate qualitative focused theory and fail to achieve quantitative results (Denscombe, 2010). Action research is useful for a continuous cycle of development and change that is geared to improving practice and resolving problems. But action research tends to involve the practitioners, particularly at the early stages (Denscombe, 2010). Practitioners are crucial people in this process and their participation has to be active, not passive (Denscombe, 2010). Hence, action research has been rejected due to limited time/resources and lack of involvement of practitioners. Surveys and sampling are excluded due to the nature of the study as data for the research will be generated by the tool proposed in this research. Mixed method applies to studies that combine alternative approaches with a single research project (Denscombe, 2010). This method generally uses both qualitative and quantitative methods (Denscombe, 2010). Our study is based on only quantitative data analysis methods. Therefore, the mixed method is not suitable for this research project.

3.2 Data Collection Method

In this study, an experiment-based data collection method has been chosen. The SLA Analyser tool which is proposed in the research, generates digital data based on the availability analysis of two cloud database services. These cloud databases were created as part of initial experiment setup. And as part of the experiment, the proposed tool generates the dataset based on the availability analysis rule as stated under the result section. This generated data is stored in the main registry database. This dataset is further utilised in the prototype for further experiment and evaluation of the SLA compliance report. The researchers focused on if the represented data is clear and unambiguous. Additionally, the research analysis was emphasized on the credibility that involves if the data is accurate (free from bias and errors).

Thereby experiment-based data collection method is selected as it aligns with the nature of aim and research questions. Questionnaires, interviews and observation methods were rejected due to inadequacy to serve the purpose of the research strategy and research question. Interviews are time consuming and more suitable for the research that require non-standardized responses (Denscombe, 2010). Using observation for collecting data was infeasible due to its demand for high resources. Moreover, a considerably time consuming personal commitment is required to observe the actual behaviour and collect the data (Denscombe, 2010). Data collection through questionnaires was not suitable due to the nature of our study as it is based on cloud availability monitoring through a proposed tool. The document-based data collection method uses documents as the main data source but in this research, the data is generated from the experiment of the study. Hence, the document-based data collection method has been rejected.

3.3 Sampling

Since the research data is digitally created during the experiment, the sampling includes all the digital data set which is the outcome of the experiments. As mentioned in the delimitation section, the results of the research are based on the data sample collected through experiments on two chosen cloud database services used by the researchers associated with the study. Thereby the data sample consists of only the availability status which were monitored for these databases. Also to be noted that the databases used are not from any practical organization, but were dataset which were created as part of the initial setup for the experiments. Though practical organizations use the service in larger terms, the availability monitoring concepts should still hold applicable.

3.4 Data Analysis Method

For this research, a quantitative data analysis method has been chosen. Quantitative research uses numbers as the unit of analysis that brings out the statistics from the collected data. These statistics are a solid foundation of description and analysis of the research results (Denscombe, 2010, p. 269).

The expected type of data obtained with the data collection method is nominal data because the data will represent the uptime/downtime of the server. There are different types of quantitative research (GCU, 2021). Statistical data analysis method has been chosen for this study. Because this research analyses to what extent server availability exists in terms of agreed-upon percentage in SLA. The research uses a one-sample proportion test (z-test) to test whether the proportion of success (uptime for our case) in a sample is significantly different from the hypothesized value (agreed upon availability percentage of SLA). It shows the achieved result from the tool is statistically significant in terms of the data. Comparison between uptime/downtime and agreed-upon availability percentage from SLA in the findings are re-assured by the statistical test.

In this study, researchers utilise a monitoring tool to analyse the database cloud service availability and establish comparison between monitored availability and availability stated in SLA. This research is not aiming to establish causal relationships or correlation between variables. Hence, other methods like correlation or experimental quantitative research methods are rejected for this study.

3.5 Ethics

Doing quantitative research requires a commitment to ethics. The researchers are required to act ethically. This implies that in order to avoid any potential ethical problems, the researcher will take into account a variety of rules (Denscombe, 2010). The authenticity and copyright of the data have been considered very carefully. No legal issues were found during the research. The equipment used are safe and personal devices of the researchers, not borrowed or taken from anywhere. Another aspect to refer to is that the researchers involved in the topic of the study are well-aware that the data collected for the research is clear and unambiguous, not fabricated or altered.

4 SLA Analyser

Service Level Agreement Analyser, also called SLA Analyser, is an availability monitoring tool. There are two parts of this tool- Core Application and Compliance Report Generator. The Core Application polls for the availability of a target database service and registers the result into the main registry database. The main registry database resides on the Core Application. The Core Application connects to a target database cloud service and checks at a defined frequency if the target database cloud service is available. The availability status of the target database service is then recorded by the Core Application into its main registry database.

The second part of the SLA Analyser tool is the Compliance Report Generator. The Compliance Report Generator calculates the availability of the target database cloud service based on the data collected in the main registry database of Core Application. Compliance Report Generator first calculates the availability of the target database cloud service using the uptime and downtime formula over a period of time. The Compliance Report Generator then compares the calculated availability against the availability parameter which is mentioned in the agreed SLA of the target database cloud service. Based on this analysis, a graphical form report is generated. This compliance report has a plot with x axis as time and y axis as availability. The SLA Analyser tool can be used to track the availability of multiple database services independent of the cloud service provider.

This research aims to utilize SLA Analyzer for independently monitoring the availability parameter of SLA for cloud based database-as-a-service. Experiment-based research strategy is employed to achieve this research aim. Two target database cloud services are monitored during the experiment using the SLA Analyser tool. At the end of the experiment SLA compliance report is generated which can be used to detect availability violations of SLA documents of the target database cloud services.

4.1 Experimental Set-up

The experimental setup of the research is demonstrated in the figure *Figure 1: Architecture of SLA Analyser*. The component of the SLA Analyser architecture is explained as under-

Core Application: Core Application consist of C# Application Program and Main Registry Database. As the name suggest, C# Application Program is the program created using C-Sharp (C#). C# is a general-purpose high-level programming language developed by Microsoft that runs on the .NET Framework. In this research C# Application Program is used to create an application which establishes connection to a target database cloud service using the database connection string. The connection string contains the parameters required for the applications to connect a database server. It includes the server instance, database name, authentication details, and some other settings to communicate with the database server. Once the connection is established, C# Application Program checks for the availability of the target database cloud service. This check is done regularly at a defined frequency. Depending upon the need for the frequency of monitoring the availability of the cloud service, the value of the frequency is configurable. The availability status of the target database service is then recorded by the C# Application Program and written into the Main Registry Database. Main Registry Database contains the database tables which store information related to the cloud service entity which are being monitored. The column names of the database table includes the availability status (Status), timestamp (Timestamp) and details of the assessed cloud database (Slave_identity_country_zone). The data is these table are filled by C# Application Program. The code base of C# Application Program is attached to Appendix A.

Compliance Report Generator: Compliance Report Generator utilises the data collected in the main registry database of Core Application to calculate the availability of the target database cloud service. The availability of the target database cloud service using the uptime and downtime formula over a period of time. The calculated availability is then compared against the availability parameter mentioned in the agreed SLA of the target database cloud service. A graphical report is generated based on this analysis.

Target Database Service: The target database cloud service is the cloud database which is being monitored by the SLA Analyser tool. For this experiment, two Target Database Services were hosted on Azure at two different availability zones. Availability zones (AZs) are isolated data centers located in specific geographical regions in which public cloud services servers are originated and operated. The information of the availability zone of the cloud service is important as the SLA of the cloud services varies based on the availability zones in which the cloud services are being hosted by the cloud providers. These Target Database Services were provisioned with fixed test data-load. This fixed data-load is read during the read operation done by the C# Application Program. Reading the same data-load ensures consistency during the read operation.

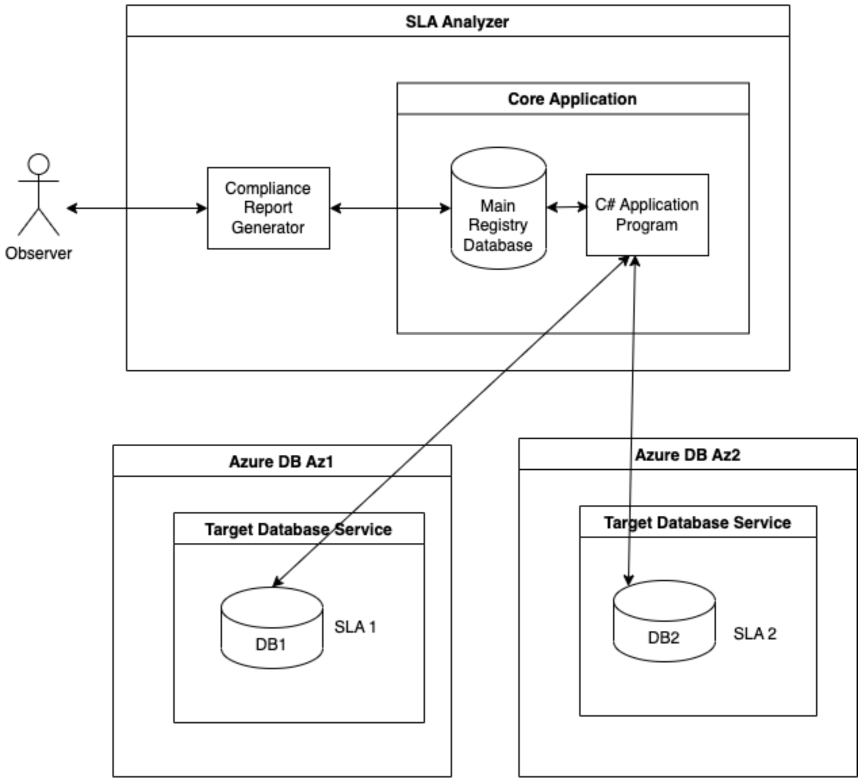


Figure 1: Architecture of SLA Analyser

4.2 Experimental Scenarios

The SLA Analyser tool was designed to be tested against certain scenarios. These scenarios ensure that the SLA Analyser accomplishes its purpose in alignment to the scope of this research (Section 1.2). Below are four vital scenarios for SLA Analyser:

> *Scenario1*: SLA Analyser tool shall monitor the available database cloud service
>
> *Scenario2*: SLA Analyser tool shall monitor the unavailable database cloud service
>
> *Scenario3*: SLA Analyser tool shall monitor more than one database cloud services
>
> *Scenario4*: SLA Analyser tool shall generate SLA compliance reports for database cloud services

The SLA Analyser tool should fulfill all the above scenarios in order to be successfully able to monitor the availability parameters of an SLA for cloud based database-as-a-services. For *Scenario1*, SLA Analyser shall be able to monitor the available database cloud services. SLA Analyser should have the ability to connect to the database cloud services and to be able to perform successful read operation on the fixed data which is stored on the database cloud services. The experimental set-up for this scenario was similar as shown in *Figure 1: Architecture of SLA Analyser.* On every successful read operation for the target database service, an entry with availability status as 'available' is stored under the main registry database.

Under *Scenario2*, SLA Analyser shall have the capability to monitor the unavailable database cloud services. SLA Analyser should monitor the situation of inability to connect or perform the read operation on the fixed data which is stored on the database cloud services. The experimental set-up for this scenario was similar as shown in *Figure 1: Architecture of SLA Analyser,* except that in the real scenario, the database services do not go unavailable so frequently by themselves. Maybe they would go down once or a few times in a month. So in order to test the unavailable database scenario, the researchers had to turn down the database server, so that the connection was lost with the database. On every unsuccessful connection for the target database service, an entry with availability status as 'unavailable' is stored under the main registry database.

For *Scenario3* SLA Analyser tool shall be able to connect to two or more database cloud services. These database cloud services can be hosted by different cloud service providers or can be on different availability zones. Each database services monitoring should be independent of each other. The connection to each database cloud services are made based on their connection string configured to the C# Application Program. The experimental set-up for connecting to two databases is shown in *Figure 1: Architecture of SLA Analyser.*

Under *Scenario4*, the dataset collected by the main registry database of Core Application of SLA Analyser should be used by the Compliance Report Generator of SLA Analyser to generate the availability plot. This plot will showcase the SLA compliance for the database cloud services. The experimental set-up for this scenario was also similar as shown in *Figure 1: Architecture of SLA Analyser.*

Besides the functional requirements, the following availability analysis rules were employed in order to decide the availability status of the database cloud services:

Rule1: If the Core Application is able to establish a successful connection to the database service, and is able to perform a successful read operation, then the database service can be marked *available*.

Rule2: If the Core Application is able to establish a successful connection to the database service, and is *not* able to perform a successful read operation, then the database service can be marked *unavailable*.

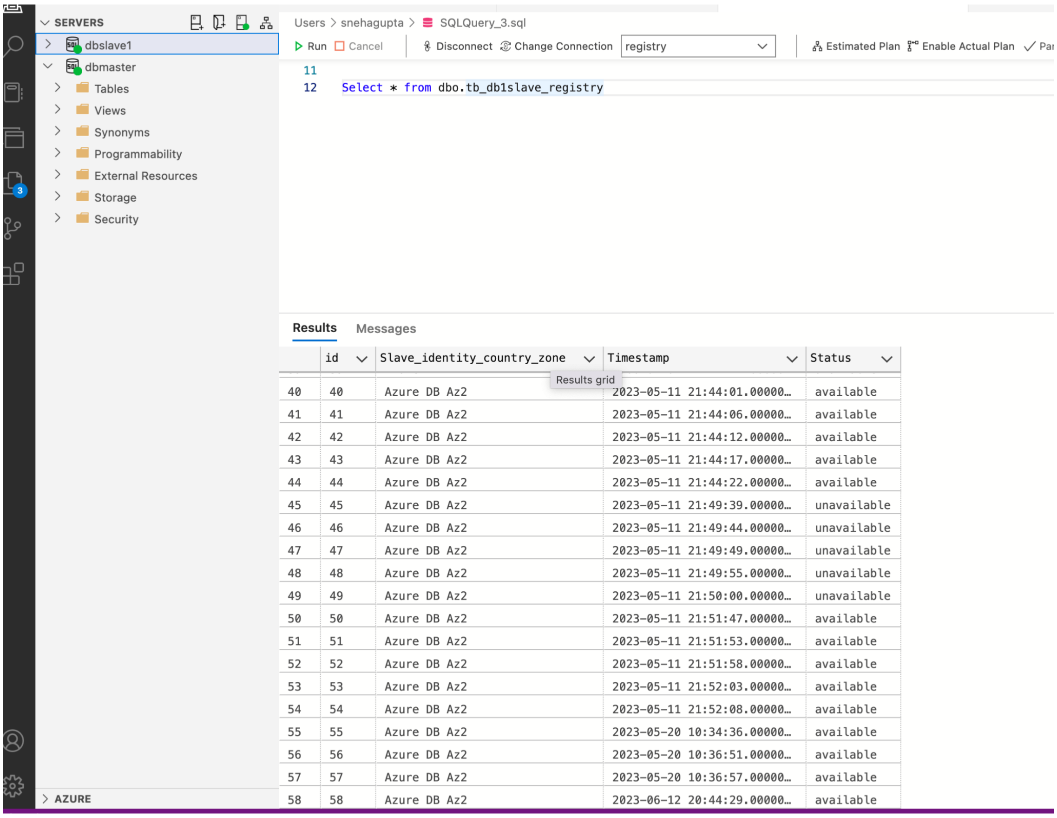
Rule3: If the Core Application is *not* able to establish a successful connection to the database service, then the database service can be marked *unavailable*.

The above stated rules were derived from the guidelines for evaluating the availability status of the cloud services. As defined in FISMA (Barker, 2003), the term 'availability' in cyber security means ensuring timely and reliable access to and use of information or services by the users. Thereby just being able to access the services does not fulfill the requirement of being available for a service. Hence *Rule2* & *Rule3* was considered *unavailable* based on this understanding. Only under condition of successful connection and on a successful read operation of the database cloud service by the tool was marked with status *available* as state in *Rule1*.

5 Result

5.1 Data Collection

During the research experiment, the SLA Analyser tool is utilized to monitor the availability parameter of SLA for cloud based database-as-a-service. This SLA Analyser tool checks the availability of the cloud services by conducting read operation on a regular frequency of time. The read operation is performed on the assessed database cloud services. The assessed database cloud services were provisioned with a test data-load which were read during the read operation done by the SLA Analyser tool. Based on the results of the read operation, the main registry database of the SLA Analyser tool was updated. The main registry database populated entries under its table which stores information related to the cloud service entity which are being monitored. The measured parameters include - the availability status (Status), timestamp (Timestamp) and details of the assessed cloud database (Slave_identity_country_zone) as shown in *Figure2 Main Registry database table*.



```
Select * from dbo.tb_db1slave_registry
```

	id	Slave_identity_country_zone	Timestamp	Status
40	40	Azure DB Az2	2023-05-11 21:44:01.00000…	available
41	41	Azure DB Az2	2023-05-11 21:44:06.00000…	available
42	42	Azure DB Az2	2023-05-11 21:44:12.00000…	available
43	43	Azure DB Az2	2023-05-11 21:44:17.00000…	available
44	44	Azure DB Az2	2023-05-11 21:44:22.00000…	available
45	45	Azure DB Az2	2023-05-11 21:49:39.00000…	unavailable
46	46	Azure DB Az2	2023-05-11 21:49:44.00000…	unavailable
47	47	Azure DB Az2	2023-05-11 21:49:49.00000…	unavailable
48	48	Azure DB Az2	2023-05-11 21:49:55.00000…	unavailable
49	49	Azure DB Az2	2023-05-11 21:50:00.00000…	unavailable
50	50	Azure DB Az2	2023-05-11 21:51:47.00000…	available
51	51	Azure DB Az2	2023-05-11 21:51:53.00000…	available
52	52	Azure DB Az2	2023-05-11 21:51:58.00000…	available
53	53	Azure DB Az2	2023-05-11 21:52:03.00000…	available
54	54	Azure DB Az2	2023-05-11 21:52:08.00000…	available
55	55	Azure DB Az2	2023-05-20 10:34:36.00000…	available
56	56	Azure DB Az2	2023-05-20 10:36:51.00000…	available
57	57	Azure DB Az2	2023-05-20 10:36:57.00000…	available
58	58	Azure DB Az2	2023-06-12 20:44:29.00000…	available

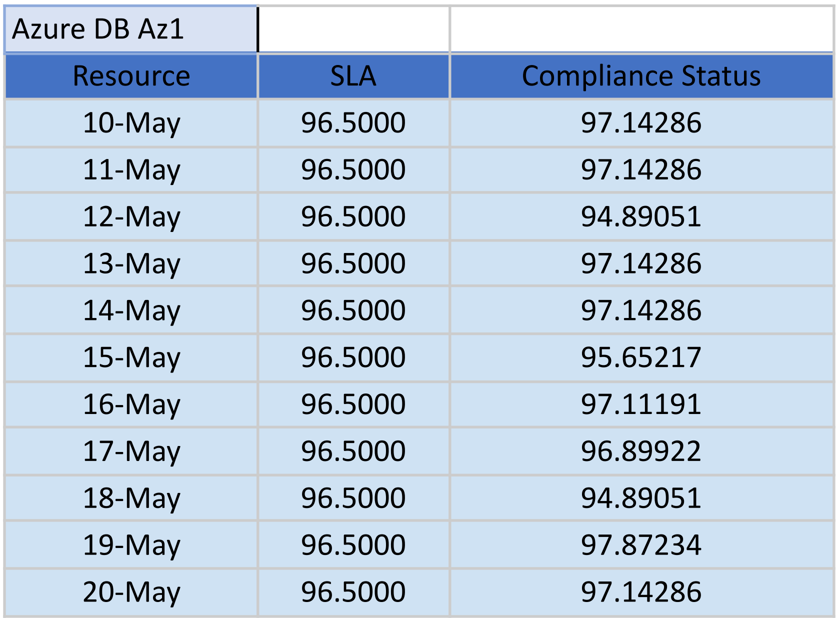
Figure 2: Main Registry database table

The Slave_identity_country_zone stored the name along with the availability zone of the cloud services which are under- accessed. The values of the availability *Status* collected were either ‘available’ or ‘unavailable’ and the time at which this was measured was stored as a *Timestamp* in the format of YYYY-MM-DD hh:mm:ss[.fractional seconds].

For this research, the time frequency at which the availability was measured was five minutes. This frequency was kept constant throughout the experiment, but its value is configurable. Depending upon the need and justified business case, a user can update the frequency of monitoring the availability of the cloud services. For customers having cloud services which are critical for their business can measure the availability of their cloud services at every second or once every minute as needed.

The availability score is a measure of percentage of the number of times a service was observed to be available during a 24 hour cycle (day). Depending upon the frequency of monitoring, for each day cycle, a considerable number of rows are inserted into the main registry database. For example, with a frequency of 5 minutes, a total of 288 rows shall be created.

The availability of each day was calculated using the uptime and downtime formula. Downtime of a service is the time during which a service is unavailable for use and the uptime of a service is the time during which a service is available for use. The availability of each day was then compared against the availability parameters in the SLA of the database cloud service. The collected data for the target database service- Azure DB Az1 is shown in *Table 1: Collected data through proposed tool.* The first column contains the dates of monitoring, the second column contains the availability parameters in the SLA of the database cloud service which was 96.5%. And the third column contains the availability of each day which was calculated using SLA Analyser. Using this data table, a compliance report was generated in the form of a graph as shown in *Figure 6: SLA Compliance Report*.



Azure DB Az1		
Resource	SLA	Compliance Status
10-May	96.5000	97.14286
11-May	96.5000	97.14286
12-May	96.5000	94.89051
13-May	96.5000	97.14286
14-May	96.5000	97.14286
15-May	96.5000	95.65217
16-May	96.5000	97.11191
17-May	96.5000	96.89922
18-May	96.5000	94.89051
19-May	96.5000	97.87234
20-May	96.5000	97.14286

Table 1: Collected data through proposed tool

5.2 Data Analysis

After the data collection of 11 days with the time-difference of 5 minutes and comparison with SLA, data analysis has been done with the help of a one-sample proportion statistical test (z-test). It evaluates if the level of success that is uptime of the cloud database (DB) services in the sample is equal or greater than the hypothesized value that is agreed upon availability percentage of SLA. The test provides insight that the tool provided result is statistically significant in terms of the data set. By analyzing with the statistical test, comparison between uptime-downtime and agreed-upon availability percentage from the tool is reevaluated and reassured.

In the beginning, data was imported in R-studio from a CSV file. After that, available/unavailable has been converted by 1/0 of the data set. For performing the one-sample proportion test following steps have been followed, i. firstly, the sample proportion($\hat{p}$) is calculated by counting the number of successes in the availability column and dividing the count by the sum of observations to get the sample proportion. ii. in this step, null hypothesis(H_0) and alternative hypothesis (H_1) are specified, such as null hypothesis is less than agreed upon availability percentage from SLA, and alternative hypothesis is equal/greater than agreed upon availability percentage from SLA, iii. standard error (SE) is calculated of the sample proportion by following the equation,

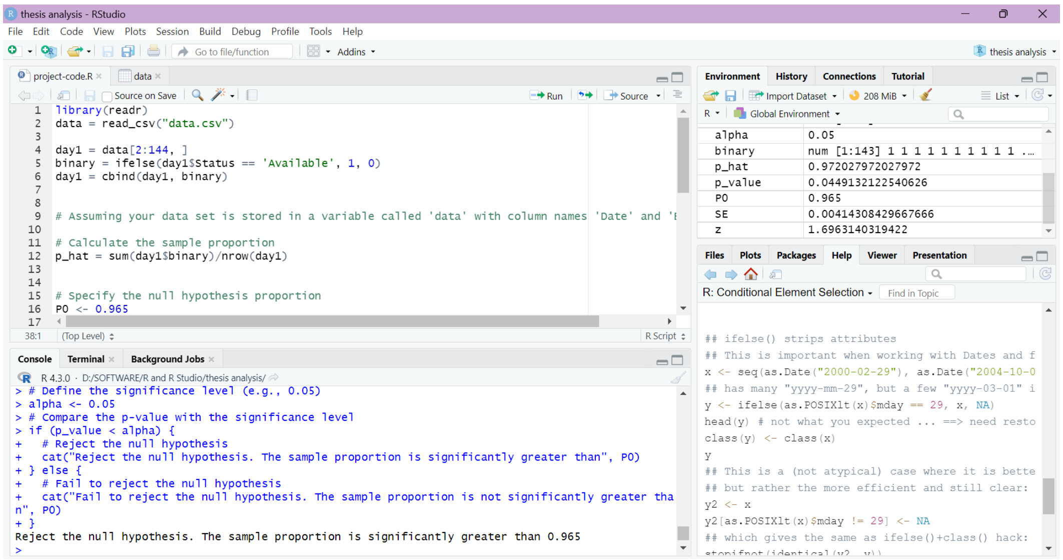
$$SE = sqrt((\hat{p} * (1 - \hat{p})) / n)$$

Here, n = sample size and $\hat{p}$ = sample proportion, iv. test statistics (z-score) value is calculated by following the equation,

$$z = (\hat{p} - P_0) / SE$$

v. p-value is calculated and compared to significance level that is 0.05. If p-value is less than significance level, null hypothesis can not be accepted.

From a total 11 days(10th May’23-20th May’23), 2 days are randomly selected for the analysis. Data from 10th May has been selected and performed on the test. The result states that the null hypothesis can not be accepted and sample proportion is significantly greater than 0.965(agreed-upon availability percentage). That explains, the actual service is available more than 96.5% in 10th May and SLA has not been breached. The *Figure 3: One-sample proportion statistical test of 10th May* below shows the results of this test.



```
library(readr)
data = read_csv("data.csv")

day1 = data[2:144, ]
binary = ifelse(day1$Status == 'Available', 1, 0)
day1 = cbind(day1, binary)


# Assuming your data set is stored in a variable called 'data' with column names 'Date' and 'E

# Calculate the sample proportion
p_hat = sum(day1$binary)/nrow(day1)


# Specify the null hypothesis proportion
P0 <- 0.965
```

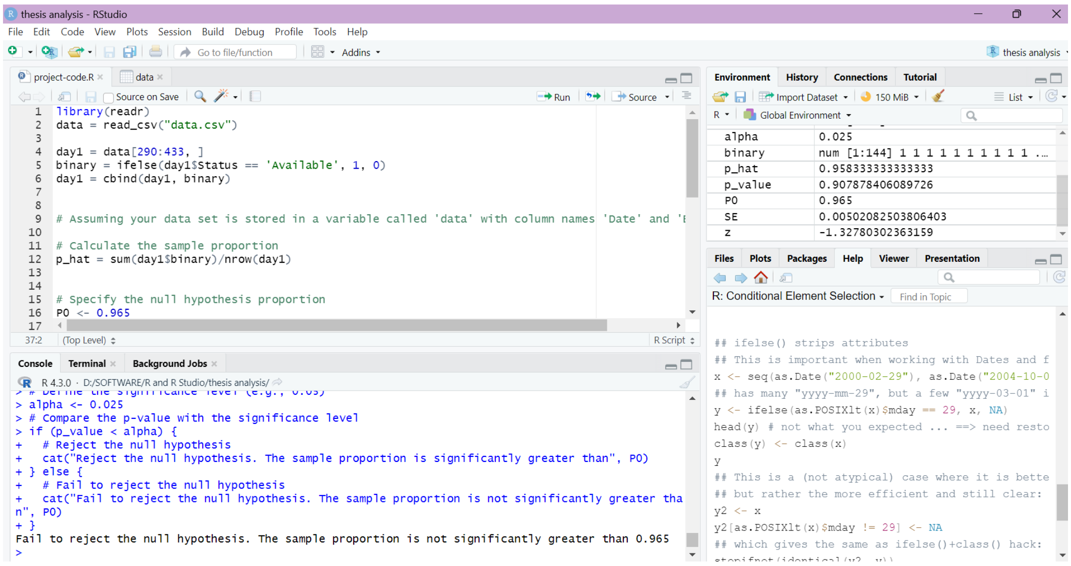
```
> # Define the significance level (e.g., 0.05)
> alpha <- 0.05
> # Compare the p-value with the significance level
> if (p_value < alpha) {
+   # Reject the null hypothesis
+   cat("Reject the null hypothesis. The sample proportion is significantly greater than", P0)
+ } else {
+   # Fail to reject the null hypothesis
+   cat("Fail to reject the null hypothesis. The sample proportion is not significantly greater tha
n", P0)
+ }
Reject the null hypothesis. The sample proportion is significantly greater than 0.965
>
```

```
alpha      0.05
binary     num [1:143] 1 1 1 1 1 1 1 1 1 1 ...
p_hat      0.972027972027972
p_value    0.0449132122540626
P0         0.965
SE         0.00414308429667666
z          1.6963140319422
```

```
## ifelse() strips attributes
## This is important when working with Dates and f
x <- seq(as.Date("2000-02-29"), as.Date("2004-10-0
## has many "yyyy-mm-29", but a few "yyyy-03-01" i
y <- ifelse(as.POSIXlt(x)$mday == 29, x, NA)
head(y) # not what you expected ... ==> need resto
class(y) <- class(x)
y
## This is a (not atypical) case where it is bette
## but rather the more efficient and still clear:
y2 <- x
y2[as.POSIXlt(x)$mday != 29] <- NA
## which gives the same as ifelse()+class() hack:
```

Figure 3: One-sample proportion statistical test of 10th May

The one-sample proportion statistical test has been done with the data of 12th May’23 as well. The result states that, null hypothesis has failed to be rejected and the sample proportion is not greater than 0.965 (agreed-upon availability percentage). That explains, the actual service has not been available 96.5% of the day of 12th May. The availability has been less than promised and SLA has been breached. The *Figure 4: One-sample proportion statistical test of 12th May* below shows the results of this test.



```r
library(readr)
data = read_csv("data.csv")

day1 = data[290:433, ]
binary = ifelse(day1$Status == 'Available', 1, 0)
day1 = cbind(day1, binary)


# Assuming your data set is stored in a variable called 'data' with column names 'Date' and 'I

# Calculate the sample proportion
p_hat = sum(day1$binary)/nrow(day1)


# Specify the null hypothesis proportion
P0 <- 0.965

```

```
> # Define the significance level (e.g., 0.05)
> alpha <- 0.025
> # Compare the p-value with the significance level
> if (p_value < alpha) {
+   # Reject the null hypothesis
+   cat("Reject the null hypothesis. The sample proportion is significantly greater than", P0)
+ } else {
+   # Fail to reject the null hypothesis
+   cat("Fail to reject the null hypothesis. The sample proportion is not significantly greater tha
n", P0)
+ }
Fail to reject the null hypothesis. The sample proportion is not significantly greater than 0.965
>
```

```
alpha      0.025
binary     num [1:144] 1 1 1 1 1 1 1 1 1 1 ...
p_hat      0.958333333333333
p_value    0.907878406089726
P0         0.965
SE         0.00502082503806403
z          -1.32780302363159
```

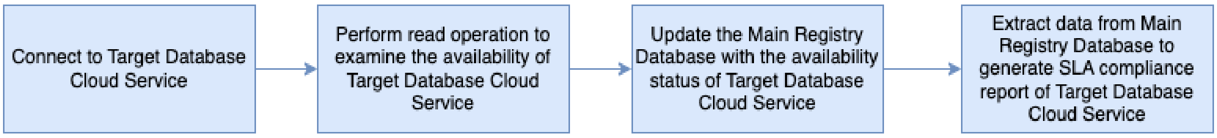
```
## ifelse() strips attributes
## This is important when working with Dates and f
x <- seq(as.Date("2000-02-29"), as.Date("2004-10-0
## has many "yyyy-mm-29", but a few "yyyy-03-01" i
y <- ifelse(as.POSIXlt(x)$mday == 29, x, NA)
head(y) # not what you expected ... ==> need resto
class(y) <- class(x)
y
## This is a (not atypical) case where it is bette
## but rather the more efficient and still clear:
y2 <- x
y2[as.POSIXlt(x)$mday != 29] <- NA
## which gives the same as ifelse()+class() hack:
stopifnot(identical(y2, y))
```

Figure 4: One-sample proportion statistical test of 12th May

By utilizing a one-sample proportion statistical test, it is reassured that the collected data through the SLA Analyser tool is statistically significant.

5.3 Findings

As per the experiment conducted in this research, the SLA Analyser tool was utilised to monitor the availability parameter of SLA for database cloud service. The connection to the target database cloud services were established by the SLA Analyser. After the successful connection was confirmed, the SLA Analyser performed a read operation on the target database cloud services. Based on the status of the read operation, the availability status of the target database cloud services was updated in the main registry database of SLA Analyser. The data from the main registry database is used to generate SLA compliance reports. The generated compliance report can be used to detect availability violations of SLA documents of the target database cloud services. This workflow of the SLA Analyser tool is also explained in *Figure 5: Workflow of SLA Analyser.*



Connect to Target Database Cloud Service
Perform read operation to examine the availability of Target Database Cloud Service
Update the Main Registry Database with the availability status of Target Database Cloud Service
Extract data from Main Registry Database to generate SLA compliance report of Target Database Cloud Service

Figure 5: Workflow of SLA Analyser

As stated before, the tool was designed based on the certain availability analysis rules. If the connection of the database was established and read operation successfully occurred, the availability status of the database service was marked as available. If the connection of the database was established and the read operation did not successfully occur, the availability status of the database service was marked as unavailable. And if the connection of the database was not established, the availability status of the database service was marked as unavailable. These stated rules were guidelines for evaluating the availability status of the target database cloud services.

Based on the availability calculation done for 11 days, a SLA compliance report graph was plotted. The x axis of the graph has the monitoring dates and the y axis of the graph has the calculated availability score. The agreed SLA which the customers and sellers agreed upon was used to check if the SLA compliance was met. *Figure 6: SLA Compliance Report* shows the SLA compliance report for the target database service- Azure DB Az1

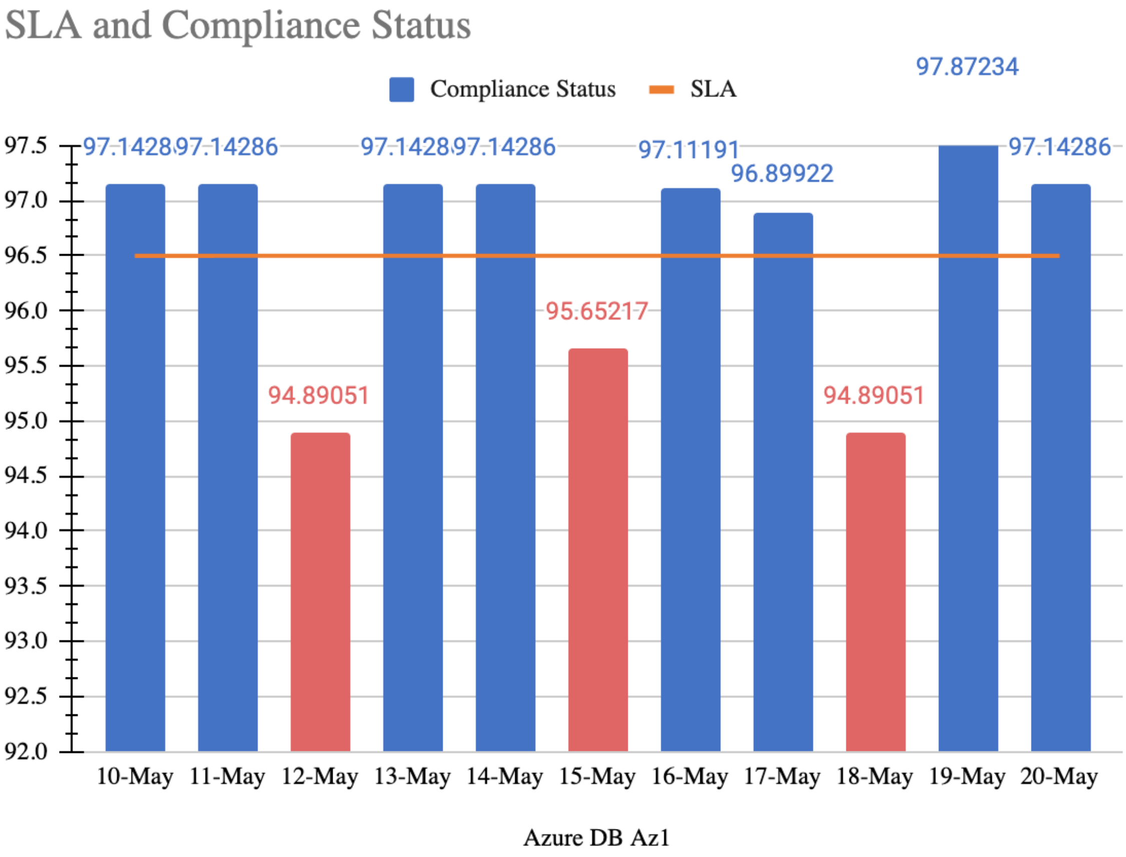


Figure 6: SLA Compliance Report

While conducting the experiment, four main experimental scenarios were created. The tool fulfilled and passed all the scenarios and was able to measure the availability parameters of an SLA for cloud based database-as-a-services. For *Scenario1*, the availability of the target database cloud services were measured based on the ability of the tool to connect to the services and to be able to perform successful read operation on the cloud services. This is a positive scenario and was able to achieve using C# programming.

The negative scenario *Scenario2* was for testing the unavailability of the target database cloud services. In the real scenario, the database services do not go unavailable so frequently by themselves. Maybe they would go down once or a few times in a month. So in order to test the unavailability scenario, the researchers had to turn down the database server, so that an "unavailable" entry is triggered. With the inability to connect to the database service raised an exception call in the tool and hence marked the service down for that timestamp.

Scenario3 was tested by connecting the SLA Analyser tool to two Azure database services of different availability zones. Each database services monitoring was independent of each other and had no manipulation on the results of the experiment. It can be stated that SLA Analyser can be used for monitoring availability for multiple cloud services.

Scenario4 was achieved by using the data set collected by the main registry database of the SLA Analyser. This data set was extracted using excel to calculate the availability of each monitoring day using the uptime and downtime formula. Based on each day's availability a plot was generated. This plot also had Availability parameters of the agreed SLA, which helped in analysing the SLA compliance for the database service. This is demonstrated in *Figure 6: SLA Compliance Report.*

6 Discussion

6.1 Discussion of the Results

The experiment and analysis provide sufficient representation for answering the research question of this paper. The proposed SLA monitoring tool significantly contributes to the field of cloud service availability and SLA compliance. Ensuring service reliability and compliance with Service Level Agreements is of utmost importance for companies that seek to maintain their competitiveness and deliver outstanding customer experience, in view of the evolution of current business practices related to Cloud computing. A number of studies have been carried out over the past years on different aspects of SLA monitoring and cloud assessment, with a view to highlighting the importance of automatic tools as well as Real Time Monitoring within this dynamic environment.

The proposed tool that is programmed with C# aims to monitor the cloud based database services by utilizing uptime and downtime of the services. The service availability and unavailability log for 11 days has been collected and compared with the SLA availability percentage through the tool. *Figure 6: SLA Compliance Report* shows that cloud service-Azure DB was not available as per the promised percentage of SLA on 12th, 15th and 18th May '23. On these days, the cloud service availability has been below 96.5% which contradicts the expected standard. On the other hand, the tool shows that the services have been available in terms of the Service Level Agreement on the other days. The proposed tool appropriately monitored the service availability with the respect to SLA. To re-evaluate the collected data from the proposed tool, one-sample proportion statistical test has been done and represented in *Figure 3* and *Figure* 4. The result of the test shows that the achieved data has been statistically significant. The SLA monitoring tool significantly minimizes the demands of both time and the needed manpower of manual monitoring by automatically evaluating SLA.

This section compares the result to the previous related studies. One notable study by Astrid Undheim, Ameen Chilwan, and Poul Heegaard titled "Differentiated Availability in Cloud Computing SLAs" centered on exploring the availability component of cloud SLAs and aimed to develop a comprehensive model for cloud data centers. While their study emphasised the fact that actual key performance indicators (KPIs) should be explicitly included in Service Level Agreements, our study has taken a further step and now actively monitors the exact KPIs set out by those agreements. This approach allows us to closely monitor whether a promised percentage of availability is actually met by cloud service providers, in order to obtain more detailed insight on actual performance and improve accountability for the fulfilment of Service Level Agreements.

Similarly, the work of Neminath Hubballi et al. in "Cloud Security Service Level Agreements: Representation and Measurement" focused on defining SLAs and proposed a method for run-time evaluation by a trusted third-party auditor. While their efforts in defining and measuring service level agreements have been valuable, our research is focused on monitoring the availability of standard service level agreements currently used between cloud providers and their customers. By doing so, our proposed tool not only ensures compliance with the SLA, but also introduces an unbiased and independent monitoring approach, eliminating any potential conflicts of interest during the monitoring process. The ability to provide independent assessments of service level agreements gives cloud operators and their clients a degree of trust and transparency, as this allows them to verify that the SLA is being properly complied with on a consistent basis.

Another relevant study conducted by Manuel Morán, Daniel Méndez, and Fabio López titled "Availability Management in Cloud Computing Based on the ISO/IEC 20000 Standard" specifically focuses on availability management in cloud computing. Their research has suggested a framework for efficient monitoring and management of cloud availability in terms of the ISO 20000 standard. Although the study contributes substantially to the management of availability, our suggested tool focuses on monitoring the availability of database-as-a-service in a cloud environment. The proposed solution enables organisations to enhance the reliability of their cloud services through improving customer satisfaction by effectively meeting with industry standards and ensuring adherence to service level agreements.

Furthermore, David Ameller and Xavier Franch present a study titled "Service Level Agreement Monitoring Tool based on measures taken from ISO/IEC 9126-1-based service oriented quality model." Their research focuses on a monitoring tool based on ISO 9126-1 standards, which is mainly aimed at quality measures. Our research focuses exclusively on the monitoring of availability and compliance with security standards for cloud based database services, complementing their work in providing valuable insight into quality measures. Our proposed tool provides organisations with a comprehensive solution to assess and optimise their database performance by emphasising the key aspect of availability, which is an integral part of the overall cloud infrastructure.

As for existing tools, cloud providers often offer monitoring tools such as CloudMonitor, ServiceWatch, and SLA Tracker, which prove to be effective in monitoring services within their respective ecosystems. These tools, which are offered by specific cloud providers, have a vital role in ensuring service responsiveness and adherence to Service Level Agreements. However, the proposed instrument provides an opportunity for unbiased and independent monitoring by ensuring that there are no potential conflicts of interest or undue influence in a monitoring process. It is particularly suited to organizations with versatile cloud environments because of its inherent flexibility and customization options. This tool offers adaptability, making it possible for businesses that use services from multiple cloud providers to monitor availability of their entire cloud environment through a single unified platform. On the other hand, tools provided by cloud vendors can be used according to one specific cloud provider. This comprehensive approach to monitoring enhances the ability of a company to manage complex cloud infrastructure efficiently.

In summary, the proposed monitoring tool presents a contribution by focusing on real time monitoring of actual key performance indicators in terms of availability for standard SLAs used between cloud providers and customers. This comprehensive monitoring approach provides valuable insights to organizations by actively monitoring the agreed-upon SLAs and database-as-a-service availability. Our experiment provides enterprises with a better understanding of how to deliver consistent, reliable cloud services in line with customer expectations. This proposed tool can also be useful for organisations that use cloud services in order to gain deeper insights on the performance of their cloud databases. This information may help them to make an informed choice of the best cloud provider according to availability insights and SLA compliance data. In addition, businesses can proactively work with their cloud services providers to address any deviations from the agreed standard service level agreements without delay by having access to real time monitoring.

Moreover, for organisations that have sensitive data or regulatory compliance requirements, an unbiased and independent approach to monitoring by the proposed tool is particularly valuable. Organisations can gain a competitive edge and help to meet their legal and regulatory obligations by demonstrating compliance with the SLAs through impartial monitoring.

Overall, the proposed SLA monitoring tool represents a significant step forward in enhancing cloud service availability monitoring capabilities. The tool enables cloud services providers and organizations to deliver proper, high quality cloud services by monitoring availability and ensuring compliance with service level agreements. The findings and contributions of this study could lead to significant improvement in cloud services reliability, customer satisfaction as well as business success at a time when cloud computing is still shaping the commercial landscape.

6.2 Result Validation

This research aims to utilise a proposed monitoring tool to validate the availability parameter of the SLA. The tool investigates if the promised SLA availability percentage is similar to actual provided service for database cloud services. Availability monitoring of database-as-a-service has been measured through the proposed tool- SLA Analyser by an experiment-based research strategy.

There are some steps taken to validate and ensure the accuracy and effectiveness of the research result. Firstly, experiment-based testing has been done to validate if the SLA analyzer collects and generates data of uptime and downtime of the database cloud services. Also, the experiment includes to verify if the system can calculate and compare between actual and SLA availability percentage properly to generate conclusions regarding if SLA is violated or not. Secondly, statistical analysis has been done to verify the results. One-sample proportion test is used to re-evaluate the findings and verify that the achieved result from the tool is statistically significant. Thirdly, comparing the proposed tool with existing monitoring systems in the market to define the contribution of the research.

This study adds value through i. independent and unbiased monitoring, ii. customizable tool in terms of customer’s business needs and, iii. offers monitoring for multiple cloud platforms with a single tool. Lastly, comprehensive literature review has been done to compare the existing tool to existing research studies related to SLA monitoring in terms of availability of cloud services. The review shows insight on the contribution and added value of this research and features the uniqueness of the proposed tool.

The research aims to determine the reliability and effectiveness of the SLA Analyzer tool for accurate measurement of cloud based Database as a Service (DBaaS) availability parameters using such validation strategies. In order to contribute to the scientific community's understanding of the monitoring of the SLA in the context of database cloud services, the experiment-based testing, statistical analysis, comparison with existing monitoring tools, and the literature review provide reliable evidence of the validity of the research findings.

6.3 Limitations

The research holds certain limitations. Firstly due to limited resources, student accounts have been used to buy cloud services from cloud providers. The student accounts have constraints on purchasing for different database sizes and offerings. Services like Power BI, which is an interactive data visualization software would have been the best fit for SLA compliance report generation but it was not integrated to the SLA Analyser tool due to limited budget. Instead excel has been used. Due to buying limitations, researchers preferred Azure cloud services, as it offered free services to students. Thereby, all the experiments for the research were conducted on Azure. However, the implementation is independent of the cloud providers and should work for any other cloud services as well.

Secondly, the proof-of-concept of the proposed tool has been performed in an artificial experimental setting. While this was the most appropriate choice for the current thesis, considering the limited time. Such settings may come with limitations regarding the generalizability and reproducibility of the results.

6.4 Future Research

A study on “the Adoption of Security SLAs in the Cloud” by Valentina Casola, Alessandra De Benedictis and Massimiliano Rak stated some issues that need to be addressed in this area. These issues can be summarized as, i. it is challenging to articulate security requirements properly, ii. it is challenging to evaluate security, and iii. security monitoring and assurance are challenging. These three points are a major obstacle to adopting security service level agreements and, indeed, cloud environments. Nonetheless, in this study, one aspect of the security that is availability has been monitored and compared with the availability-percentage from SLA.

This study builds up the foundation for future research to further expand the tool to integrate parameters on the top of availability to pursue a better comprehensive monitoring system for cloud services. To further expand the research, monitoring the other aspects of security that are confidentiality and integrity can be explored. Additionally, future research can include how to specify the actual KPI that defines confidentiality and integrity in SLA. With the help of those KPI, the confidentiality and integrity can be measured and monitored. Furthermore, this study focused on the database-as-a-service model, future study can explore different cloud service models such as, infrastructure-as-a-service, platform-as-a-service, in terms of SLA monitoring. Future research can build upon the findings of this study to battle the limitations, expand the tool and explore different aspects of cloud service monitoring for example- Containers, IOT devices, Servers, etc.

6.5 Conclusion

In conclusion, the aim of this study was to utilize the SLA Analyzer tool, monitoring tool to independently monitor the availability parameter of SLA for cloud based database-as-a-service. The research aim was accomplished by performing an experiment with certain scenarios. The paper highlighted the importance of independent monitoring of cloud services to ensure adherence to SLA commitment. The findings shed light on the aim of this study and provided reliable results of monitoring SLA agreements. The proposed tool showed through the experiment, there were some days that SLA percentage did not match with the service availability. It indicated that the actual services were not provided as promised and SLA was breached. The problem addressed in this paper is general and significant interest of the customers as the companies and organizations significantly adopting cloud services and facing challenges to monitor service availability. Due to the small-scale nature of this work, the contributions and implication for the scientific community was limited to addressing the particular problem of cloud service availability monitoring through a tool that validates the actual service with promised SLA. This study offered a foundation for future studies to explore further aspects of SLA management of cloud computing that can benefit both customer and cloud service providers.

References

Alhamazani, K. *et al.* (2014) “An overview of the Commercial Cloud Monitoring Tools: Research Dimensions, design issues, and state-of-the-art,” *Computing*, 97(4), pp. 357–377. Available at: https://doi.org/10.1007/s00607-014-0398-5.

Badger, M.L. *et al.* (2012) “8.3.2 Service Agreement Evaluation,” in *Cloud computing synopsis and recommendations*. Gaithersburg, MD: U.S. Dept. of Commerce, National Institute of Standards and Technology.

Barker, W.C. (2003) *Guideline for identifying an information system as a national ... - NIST, NIST SP 800-59*. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-59.pdf (Accessed: 30 July 2023).

Barros, A. (2015) “rSLA: Monitoring SLAs in Dynamic Service Environments,” in *Service-oriented computing: 13th International Conference, ICSOC 2015, Goa, India, November 16-19, 2015: Proceedings*. Heidelberg: Springer.

Begic, K. and Tanovic, A. (2012) ‘Improvement of implementation of ISO-IEC 20000 edition 2 standard in IT systems of telecom operator through comparison with Itil V3 Best Practices’, *2012 20th Telecommunications Forum (TELFOR)* [Preprint]. doi:10.1109/telfor.2012.6419140.

Capel, M., Aporta, O. and Pegalajar-Jiménez, M. (2020) “Quality of service in cloud computing environments with Multitenant DBMS,” *Proceedings of the 10th International Conference on Cloud Computing and Services Science* [Preprint]. Available at: https://doi.org/10.5220/0009794605060514.

Casola, V., De Benedictis, A. and Rak, M. (2015) “On the adoption of security slas in the cloud,” *Lecture Notes in Computer Science*, pp. 45–62. Available at: https://doi.org/10.1007/978-3-319-17199-9_2.

Casola, V., De Benedictis, A. and Rak, M. (2015) ‘Security monitoring in the cloud: An SLA-based approach’, *2015 10th International Conference on Availability, Reliability and Security* [Preprint]. doi:10.1109/ares.2015.74.

Cérin C., *et al.* (2013) *Downtime statistics of current Cloud Solutions - iwgcr.org*, *Downtime statistics of current cloud solutions. International Working Group on Cloud Computing Resiliency, Tech. Rep.* . Available at: http://www.iwgcr.org/wp-content/uploads/2013/06/IWGCR-Paris.Ranking-003.2-en.pdf (Accessed: 27 July 2023).

Cloud migration, computing, and more Zippia 25 Amazing Cloud Adoption Statistics 2023 Cloud Migration Computing And More Comments. Available at: https://www.zippia.com/advice/cloud-adoption-statistics (Accessed: April 2, 2023).

Dana, P. (2014) “A taxonomy for sla-based monitoring of cloud security,” in *2014 IEEE 38th Annual computer software and applications conference*. IEEE, pp. 640–641.

Denscombe, M. (2010) The good research guide: For small-scale social research projects (4ed) Berkshire: Open University Press

Endo, P.T. *et al.* (2016) ‘High availability in clouds: Systematic review and Research Challenges’, *Journal of Cloud Computing*, 5(1). doi:10.1186/s13677-016-0066-8.

Faction Inc. and Amanda (2022) *Multi-Cloud Trends: The latest trends of 2022*, *Faction Inc.* Available at: https://www.factioninc.com/blog/hybrid-multi-cloud/multi-cloud-trends/ (Accessed: April 2, 2023).

Fan, J., Ammar, M. H., & Zegura, E. W. (2015). MonSLAR: Automated monitoring of SLAs for cloud services. IEEE Transactions on Network and Service Management, 12(4), 513-526. doi:10.1109/TNSM.2015.2486283

Frey, S., Reich, C. and Lüthje, C. (2013) “Key performance indicators for cloud computing SLAs.,” *In The fifth international conference on emerging network intelligence, EMERGING*, pp. 60–64.

Hubballi, N. *et al.* (2019) “Cloud Security Service Level Agreements: Representation and measurement,” *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* [Preprint]. Available at: https://doi.org/10.1109/infcomw.2019.8845105.

Januzaj, Y., Ajdari, J. and Selimi, B. (2015) “DBMS as a cloud service: Advantages and disadvantages,” *Procedia - Social and Behavioral Sciences*, 195, pp. 1851–1859. Available at: https://doi.org/10.1016/j.sbspro.2015.06.412.

Johannesson, P. and Perjons, E. (2014) “An introduction to design science” Available at: https://doi.org/10.1007/978-3-319-10632-8.

Kaaniche, N. *et al.* (2017) “Security SLA based monitoring in clouds,” *2017 IEEE International Conference on Edge Computing (EDGE)* [Preprint]. Available at: https://doi.org/10.1109/ieee.edge.2017.20.

Kosinski, J. *et al.* (2008) ‘SLA monitoring and Management Framework for telecommunication services’, *Fourth International Conference on Networking and Services (icns 2008)* [Preprint]. doi:10.1109/icns.2008.31.

Kuć, R. (2023) *20 Best Cloud Monitoring Tools & Services [2023 comparison]*, *Sematext*. Sematext. Available at: https://sematext.com/blog/cloud-monitoring-tools/ (Accessed: February 16, 2023).

Manuel, P. (2013) “A trust model of cloud computing based on quality of service,” *Annals of Operations Research*, 233(1), pp. 281–292. Available at: https://doi.org/10.1007/s10479-013-1380-x.

Oliveto, F.E. (1999) ‘An algorithm to partition the operational availability parts of an optimal provisioning strategy’, *Annual Reliability and Maintainability. Symposium. 1999 Proceedings (Cat. No.99CH36283)* [Preprint]. doi:10.1109/rams.1999.744137.

Quantitative research design methods for writing dissertations (2021) *GCU*. Available at: https://www.gcu.edu/blog/doctoral-journey/quantitative-research-design-methods-writing-dissertations (Accessed: April 3, 2023).

Rosencrance, L., Louissaint, S. and Brush, K. (2021) *What is a service-level agreement (SLA)?*, *IT Channel*. Available at: https://www.techtarget.com/searchitchannel/definition/service-level-agreement (Accessed: 27 July 2023).

Sousa, F.R. *et al.* (2012) "Quality of Service for Database in the Cloud.," *Quality of Service for Database in the Cloud.* , pp. 595–601.

Taha, A. *et al.* (2020) "Decentralized runtime monitoring approach relying on the Ethereum Blockchain Infrastructure," *2020 IEEE International Conference on Cloud Engineering (IC2E)* [Preprint]. Available at: https://doi.org/10.1109/ic2e48712.2020.00021.

Toeroe, M. and Tam, F. (2012) *Service availability: Principles and practice*, *Wiley.com*. Available at: http://www.wiley.com/WileyCDA/WileyTitle/productCd- 1119954088.html (Accessed: 28 July 2023).

Undheim, A., Chilwan, A. and Heegaard, P. (2011) "Differentiated availability in cloud computing slas," *2011 IEEE/ACM 12th International Conference on Grid Computing* [Preprint]. Available at: https://doi.org/10.1109/grid.2011.25.

Wang, S., Xu, C., Chen, X., & Liu, A. (2018). MCSLAMon: A Multi-Cloud SLA Monitoring System. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (pp. 1094-1099). IEEE. doi:10.1109/COMPSAC.2018.00151

What is a Service Level Agreement (SLA)? (2021) *PagerDuty*. Available at: https://www.pagerduty.com/resources/learn/what-is-service-level-agreement/ (Accessed: April 2, 2023).

Zawoad, S., & Thool, R. C. (2021). SLAMON: An Automated Monitoring Framework for Cloud Services SLA Compliance. In 2021 IEEE International Conference on Services Computing (SCC) (pp. 12-19). IEEE. doi:10.1109/SCC52569.2021.00009

Appendix A – SLA Analyser code

Program.cs

```
using sladoctor.DAL;
using Microsoft.Extensions.Configuration;
using System;
using System.IO;
namespace sladoctor
{
    class Program
    {
        private static IConfiguration _iconfiguration;
        static void Main(string[] args)
        {
            int FREQUENCY = 300000; //Frequency of monitoring. Wait 5mins
            int AzureDBAz1 = 1;
            int AzureDBAz2 = 2;
            GetAppSettingsFile();
            //Continuous monitoring with desired FREQUENCY
            while(true)
            {
                Thread.Sleep(FREQUENCY); //runs as per desired frequency
                CheckAvailabilityOfDatabase(AzureDBAz1);
                CheckAvailabilityOfDatabase(AzureDBAz2);
            }
        }
        static void GetAppSettingsFile()
        {
            var builder = new ConfigurationBuilder()
                                .SetBasePath(Directory.GetCurrentDirectory())
                                .AddJsonFile("appsettings.json", optional: false, reloadOnChange: true);
            _iconfiguration = builder.Build();
        }
        //checks availability of database by reading data from a table
        static void CheckAvailabilityOfDatabase(int AzureDBID)
        {
            string param1 = "Azure DB Az"+AzureDBID;
            DateTime param2 = DateTime.UtcNow;
            string param3;
            var countryDAL = new CountryDAL(_iconfiguration);
            var listCountryModel = countryDAL.GetList(AzureDBID);
            if (listCountryModel.Count > 0)
            {
                Console.WriteLine(param1+" :Slave is Up!");
                param3 = "available";
            }  else {
                Console.WriteLine(param1 + " :Slave is down!");
                param3 = "unavailable";
            }
            //on successful read operation, writes the status to a registry database
            countryDAL.updateRegistry(param1,param2.ToString(),param3);
        }
    }
}
```

CountryDAL.cs

```
using sladoctor.Model;
using Microsoft.Extensions.Configuration;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System;
namespace sladoctor.DAL
{

            public class CountryDAL
            {
```

```
        private string _connectionString;
        private string _connectionStringDB1;//for azure slave in Az1
        private string _connectionStringDB2;//for azure slave in Az2
        private string _connectionStringRegistry;// for registry
        public CountryDAL(IConfiguration iconfiguration)
        {
            _connectionStringDB1 = iconfiguration.GetConnectionString("AzureDB1");
            _connectionStringDB2 = iconfiguration.GetConnectionString("AzureDB2");
            _connectionStringRegistry = iconfiguration.GetConnectionString("Registry");
        }

        //by executing a read operation, check if the slave is running
        public List<CountryModel> GetList(int DBNumber)
        {
            var listCountryModel = new List<CountryModel>();
            try
            {
                //Console.WriteLine(DBNumber);
                if (DBNumber == 1)
                    _connectionString = _connectionStringDB1;
                else if (DBNumber == 2)
                    _connectionString = _connectionStringDB2;

                using (SqlConnection con = new SqlConnection(_connectionString))
                {
                    SqlCommand cmd = new SqlCommand("SP_COUNTRY_GET_LIST", con);
                    cmd.CommandType = CommandType.StoredProcedure;
                    con.Open();
                    SqlDataReader rdr = cmd.ExecuteReader();
                    while (rdr.Read())
                    {
                        listCountryModel.Add(new CountryModel
                        {
                            Id = Convert.ToInt32(rdr[0]),
                            Country = rdr[1].ToString(),
                            Active = Convert.ToBoolean(rdr[2])
                        });
                    }
                }
            }
            catch (Exception ex)
            {
                throw;
            }
            return listCountryModel;
        }

        public bool updateRegistry(string arg1, string arg2, string arg3) {
            try
            {
                using (SqlConnection con = new SqlConnection(_connectionStringRegistry))
                {
                    //SqlCommand cmd = new SqlCommand("INSERT INTO dbo.tb_db1slave_registry VALUES
(\'sample_se_az11\',CURRENT_TIMESTAMP, \'available\')", con);
                    SqlCommand cmd = new SqlCommand("INSERT INTO dbo.tb_db1slave_registry VALUES (\'"
+ arg1 + "\',\'" + arg2 +"\',\'" + arg3 + "\')", con);
                    cmd.CommandType = CommandType.Text;
                    con.Open();
                    int rowsadded = cmd.ExecuteNonQuery();
```

```
               if (rowsadded > 0)
               {
                  Console.WriteLine("Rows added to registry");
                  return true;
               }
               else
               {
                  Console.WriteLine("No row inserted to registry!");
                  return false;
               }

            }
         }
         catch (Exception ex)
         {
            throw ex;
         }
      }
   }
}
```

Appendix B – Reflection Document 1

By Mosammath Nazifa Anjum

This section aims to reflect on the research study in terms of goals, planning, relevance and satisfaction of the work.

The study correspondence to the thesis goal that is proposing a tool for monitoring availability of database-as-a-service in terms of SLA. An experiment has been done to validate if the proposed tool can successfully collect and generate uptime/downtime of the cloud services and after that, the tool can compare between actual availability and agreed upon availability percentage of SLA. The tool is able to provide insight when the SLA is breached. The plan initially started with exploring the topic and narrowing down the goal to balance with the time and resources of current study. A plan of the study has been made by following the steps of the thesis template such as, problem finding, literature review, implementation, results and discussion. The plan was effective and successful to develop the tool and validate it through experiment. Additionally, detailed project planning could help the researchers to improve the study even more. Moreover, the work is related to my programme 'Information Security'. Availability is an essential part of CIA(confidentiality, integrity, availability) that is the foundation of security. This study focuses on availability of cloud services. 'Information Security in Organizations' and 'Database Management Systems' are two most relevant for the thesis work. The thesis contributes significant value to scientific studies and future work. Businesses and organizations are highly adopting cloud-computing for easy to use and flexibility. Therefore, the SLA based monitoring system plays an important role and future research can expand with other aspects of security for a comprehensive monitoring tool. Nonetheless, I am satisfied with overall thesis work and results. SLA Analyzer has been developed successfully that can validate the SLA to verify that promised availability of the services are provided in terms of SLA. Additionally, the findings are validated with a one-sample proportion test that shows the data achieved through the tool are statistically significant and accurate. In conclusion, the study effectively addressed the problem and fulfilled the aim and goals of the research.

Appendix B – Reflection Document 2

By Sneha Gupta

● How does your study correspond to the goals of the thesis course? Why? Focus on the goals that were achieved especially well and those that were not well achieved.
-The study- Master in Information Security corresponds to courses which build a solid foundation for this thesis. The understanding on topics of security and importance of availability formed the basis of this thesis. The study also introduced us to the cloud security and monitoring topics. Goal of the thesis was well achieved in terms of learning the concepts of availability, cloud security, SLA, monitoring tools, etc. Due to certain constraints, we could not extend the thesis to the point we wanted to. We initially wanted to test the proposed tool with different cloud providers, but due to limitations of student accounts budget we could afford those services. Had we been working on this thesis in an industrial setup, we would have been able to improve the thesis.

● How did the planning of your study work? What could you have done better?
-The planning of the study worked fine. Firstly we tried to understand all the milestones of the thesis and tried to time bound all the activities related to each milestone. Since I was working on the thesis with a partner, clear communication and collaboration were the key to success. We tried to divide every task into halves, so that both of us get equal opportunity to learn and demonstrate. However we could have done a little better in one area. We were a bit behind our track during the implementation phase of the tool. We started implementing two weeks late as we were waiting for the results for Phase 2 Rough draft approved by the reviewer. It was a bit of a mistake from our end in our planning as the results for Phase 2 Rough draft approved by the reviewer had no implication on the implementation plan of the tool. Thereby we took time for implementation and had to rush in chapter 4 & 5 in order to finish the thesis on time. Overall with the help of our supervisor, I personally believe we could finish our thesis on time with satisfactory results.

● How does the thesis work relate to your education? Which courses and areas have been most relevant for your thesis work?
-The thesis work directly relates to one of the pillars of the security triad- Availability. Courses like Introduction to Information Security (INTROSEC) and Cyber Security (CYBER) have been relevant to the thesis work as it introduces cloud security topics.

● How valuable is the thesis for your future work and/or studies?
-The thesis gave me an opportunity to learn about cloud services and about the security aspects of cloud services. I want to pursue my career as a Cloud Security Engineer and this thesis was my first stepping stone towards my future work. In the future, I would like to work with the Security Operation center(SOC) team in the IT industry, where I would like to learn and contribute to the security monitoring tools and activities.

● How satisfied are you with your thesis work and its results? Why?
-Throughout this thesis work, I learned a lot about cloud security. I can across research papers and articles which give valuable insight in the domain of cloud engineering. Personally, I am satisfied with this thesis work and its results as the thesis tries to solve one of the challenges faced in the industry while using cloud services. The result helps in monitoring availability of cloud services and validating it against the SLA. The proposed tool is unbiased and helps monitor multiple cloud services using the same solution.