National College of Ireland

Designing and Implementing a Comprehensive Cloud Security Monitoring Tool with CloudWatch Logs and CloudWatch Console

MSc Research Project
Cloud Computing

Pranav Patil
Student ID: 23193867

School of Computing
National College of Ireland

Supervisor:Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Pranav Rajendra Patil
Student ID:	23193867
Programme:	Cloud Computing
Year:	2024
Module:	Msc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	12/12/2024
Project Title:	Designing and Implementing a Comprehensive Cloud Security Monitoring Tool with CloudWatch Logs and CloudWatch Console
Word Count:	7097
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Pranav Patil
Date:	12/12/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Designing and Implementing a Comprehensive Cloud Security Monitoring Tool with CloudWatch Logs and CloudWatch Console

Pranav Patil
Student ID: 23193867

Abstract

The reliability, performance, and safety of cloud-based systems can be ensured through cloud security monitoring. This research discusses the design and implementation of a comprehensive cloud security monitoring tool based on AWS-native services, particularly CloudWatch Logs and CloudWatch Console. Motivation for this work came from the need for real-time monitoring solutions that are designed to seamlessly integrate with cloud environments in order to offer efficient log collection, performance tracking, and troubleshooting capabilities. The project included provisioning an Amazon EC2 instance, configuring it with the CloudWatch Agent to collect both system logs and metrics. A CloudWatch log group monitorloggroupec22024 had the logs streamed from critical system activities such as initialization and resource usage. CloudWatch Metrics monitors CPU, memory, and disk utilization in real time. These logs and metrics would primarily be visualized from a central place, the CloudWatch Console, instead of using some third-party tools like Elasticsearch or Kibana. During implementing the solution, there have been a few permission errors caused by IAM roles and configuration validation errors. Those issues were iteratively debugged and solved with the policy change. The final system would create a seamless pipeline of log collection, processing, and visualization, thus proving that the AWS-native tools could be used effectively in security monitoring. This research emphasizes the ease, cost-effectiveness, and scalability of using CloudWatch for comprehensive monitoring solutions in cloud environments. It concludes by suggesting potential future enhancements, including automated alerting and advanced integrations with third-party tools for further analysis. This work focuses much on the practicality of monitoring for modern IT infrastructure.

1 Introduction

Cloud computing has brought the way business and organization use to scale their IT infrastructure by revolutionizing its management. Platforms such as AWS give high flexibility, scalability, and affordability, but bring out several issues in security and performance monitoring. This calls for greater attention on cloud monitoring systems to gain real-time visibility into cloud-based resource operations. These systems essentially allow them to find vulnerability, avoid downtime, and comply with regulatory standards. This paper focuses on the development and implementation of a holistic cloud security

monitoring tool using AWS-native services, such as CloudWatch Logs and CloudWatch Console. In contrast to the traditional approaches that make use of other tools, such as ELK Stack, which is made up of Elasticsearch, Logstash, and Kibana, it does show how perfectly log collection, performance monitoring, and security analytics are made achievable with native cloud tools. This project works through the launch and configuration of an EC2 instance, ties the instance with CloudWatch Agent, and further analyzes log systems and metrics through CloudWatch Console. This chapter introduces the problem context, outlines the research question that drives this study, and explains the motivations for a cloud-native monitoring solution. Then, ethical considerations in relation to data privacy and resource management are discussed, which forms the background for the following sections.

1.1 Research Question

How can one leverage AWS-native tools in the design and implementation of an affordable, scalable, and efficient cloud security monitoring solution?
This question will address the need for affordability as well as efficiency in new generations of cloud monitoring systems. Traditional reliance on tools external to AWS often gives rise to added complexity and operational overhead, making deployment undesirable in SMEs. Focusing on tools native to AWS, the research should naturally simplify the monitoring process while taking advantage of AWS's integration benefits.

1.2 Motivation

This project is motivated by the growing dependence of organizations on cloud services for their critical operations. With growing complexity in cloud environments, maintaining security, performance, and reliability becomes increasingly challenging. The ELK Stack, with its superior indexing and visualization capabilities, remains one of the preferred monitoring systems. However, it comes with resource-intensive overheads, the need for external integrations, and complex setups that pose barriers to organizations with limited technical expertise or budgets.
Native AWS tools such as CloudWatch provide an integrated alternative that is native in the AWS ecosystem. These provide:

- **Ease of Use**: No need to set up or maintain monitoring servers from the outside.
- **Cost-Effective**: AWS services are based on a pay-as-you-go model, meaning no initial investments.
- **Scalability**: CloudWatch can handle massive cloud deployments without much configuration change.
- **Seamless Integration:** Because CloudWatch is part of the AWS environment, it integrates directly with other services like EC2, S3.

Trends to AWS-native solutions further align with industry trends of simple, automated, and integrated services. Focusing on CloudWatch Logs and CloudWatch Console, this research has shown how it may meet monitoring needs while still providing a foundation for possible additions such as alert automation and third-party integrations.

The increasing need to monitor real-time logs and metrics to respond to security threats is another motivating factor. Advanced cyberattacks demand that organizations have a system in place to detect anomalies or unauthorized access to resources immediately. This research demonstrates the immediate insight that CloudWatch Logs can give into the behavior of the system, thereby enabling proactive responses against potential threats.

2 Related Work

2.1 Cloud Monitoring Tools and Frameworks

- Monitoring of cloud resources, therefore, involves tracking performance, availability, and security of cloud infrastructure. Ahola, (2022) maintains that an all-rounded monitoring solution in clouds such as Dynatrace allows real-time visibility into cloud environments thus identifying and fixing problems proactively [1]. Similarly, Diagboya (2021) discussed the capabilities of Amazon CloudWatch in monitoring AWS infrastructure, such as optimization of resource allocation, detection of anomalies, and automation of response to predefined conditions [2]. Such tools provide various functionalities, such as metrics collection, log management, and alerting mechanisms, which are essential for maintaining cloud system health and security.

- Firdhous et al. (2015) assess several providers of cloud systems, placing a special emphasis on models, methods, and approaches applied to monitoring. Therefore, their analysis underlines an appropriate choice of monitoring structures appropriate for the needs and requirements of an organization to maintain security [3]. Finally, Guide (2009) describes Amazon CloudWatch in detail, describing feature characteristics and integration into an ecosystem of AWS [4]. These studies collectively summarize the diverse landscape of cloud monitoring tools and the factors that critically influence their efficiency in different cloud environments.

- Fatema et al. (2014) provide a comprehensive review of cloud monitoring tools, thus presenting a taxonomy that can categorize tools based on the capabilities and objectives [16]. It helps in selecting appropriate monitoring solutions for the organizations by explaining the functionalities and limitations of various tools. Stephen, Benedict, and Kumar, (2019) explored multiple cloud monitors for an IaaS environment. Therefore, multiple tools can complete each other in order to present a more holistic view of the system [12] [18]. The key takeaway here is that when utilizing a number of these solutions it enhances visibility to help create an overall better posture in regards to security and safety within cloud infrastructures.

2.2 Security Monitoring in Cloud Environments

- Security monitoring in the cloud is the continuous observation of analysis of system activities about threats and their possible ways of response. According to Kozlovszky (2016) and his views, emphasis was given on the fact of necessity of proactive security risks management and mitigation [14]. The study by Tedeschi et al. (2015) focuses on the security aspect of cloud-based condition monitoring for machine tools: "Challenges and Solutions Associated with Securing Industrial

Cloud Applications" [15]. Such studies demonstrate the myriad concerns over security in the cloud and the monitoring process as a crucial enabler for dealing with these issues.

- Murthy, Siddesh, and Srinivasa (2024) provides a comprehensive overview of cloud security concepts, applications, and practices, providing an insight into the strategies and technologies used to secure the cloud infrastructures [7]. Their work emphasizes the integration of security monitoring tools such as CloudWatch Logs and the CloudWatch Console to increase detection and mitigation of security incidents. Further, Nadon and Nadon (2017) explain logging and monitoring of websites that move to AWS and provide effective security and reliability of applications running on the cloud [8].

- Routavaara (2020) emphasized on AWS public cloud security monitoring while using the tools and best practices to detect and handle security issues [11]. This emphasis on AWS-specific security monitoring is relevant to the research topic, focusing on practical uses of CloudWatch services in real-world applications. Moreover, Stephen et al. (2019) study different cloud monitors for IaaS, comparing their performance in different security contexts [12][18]. The strengths and weaknesses of various monitoring tools are determined by their analysis, thus making the right choice of the solution according to the particular security needs.

2.3 Log Management and Analysis

- The most fundamental elements in cloud security monitoring include effective log management and analysis, whereby an organization tracks the activities going on within its system, discovers anomalies, and complies with all the regulatory requirements. Quézet (2016) researches log files and their application in the proactive monitoring of big data environments; however, he underlines that a more complete log analysis can pinpoint security threats [5]. His research concerns himself with the problems of dealing with enormous amounts of log data and the reasons why advanced analytics is a necessity in deriving insights.

- Nikkhouy (2016) discussed monitoring service chains in the cloud, focusing primarily on the interaction between different services and how cohesive log management is essential [9]. This research supports the integration of log management solutions that correlate events across service boundaries to offer a holistic understanding of the cloud environment. Penberthy and Roberts (2022) discussed monitoring and observability in the context of a deployment of applications on Amazon Web Services, best log management practices, and ways to enhance visibility and traceability through CloudWatch Logs [20].

- With an emphasis on practical application to everyday development on the AWS Cloud using Python, Sakinmaz (2023) even guides developers on how one goes about deploying and managing cloud applications [21]. His document highlights strategies on how to optimize the use of CloudWatch Logs with access to the CloudWatch Console on effective log management and evaluation within real applications. Furthermore, Nousiainen (2020) emphasizes the role of observability in improving cloud

governance and advocates for more visibility through the best practices of log management [10][22]. His study shows how increased observability can lead to better governance and security outcomes in cloud environments.

- Ibrahim et al. (2011) introduced Cloudsec, which is a security monitoring appliance for virtual machines in the IaaS cloud model [17]. The appliance relies on log data to monitor and secure virtual machines. This is another representation of log-based security monitoring in cloud infrastructures. This would go in line with the research focus on the usage of CloudWatch Logs for overall security monitoring, showing that log-based solutions are efficient in enhancing security in the cloud.

2.4 Best Practices and Future Directions

- Best practice following the trend and technological progress regarding this is observed in efficient cloud security monitoring. The paradigm known as "Security as Code," as stated by Das and Chu (2023), "brings security practice into the development and deployment process through automation and codification" [6]. This fits well with the use of CloudWatch Logs and CloudWatch Console insofar as this makes automatic monitoring and management of events inside the cloud environment possible.

- Mounika (n.d.) offers the integration of Nginx Web Server with AWS CloudWatch Monitoring Service: configurations and optimization in practical perspective toward monitoring web applications [19]. This case study puts more emphasis on how best configurations of monitoring need to be tailored toward the design architecture of specific applications if excellent security and performance results have to be obtained. Routavaara, again in 2020 states there is a need to work continuously on improving monitoring strategies, and the call up for the use of machine learning algorithms and deep analyses in order to effectively work towards threat detection [11].

- Verginadis (2023) reviewed several probes for cloud continuum monitoring and their cases for use as well as the efficiency in various clouds, [13]. Analysis from the findings reveals areas of innovation needed in the monitoring of the cloud to fill these gaps. These areas can present pathways for further research and development. Such an outlook keeps pace with cloud security monitoring because evolving technologies and methods keep challenging what is known and progress further improvements in security.

- Future directions in cloud security monitoring lie in the direction of bringing in artificial intelligence and machine learning for automated threat detection and response, wide-scale adoption of standardized monitoring frameworks to ensure interoperability and consistency across monitoring entities, and enhancing log management to deal with overwhelming volume and complexity of logs. Since cloud environments are increasing their heterogeneity and dispersion, there is an increasingly required unifying solution of monitors that would offer all-rounded views across platforms and services.

3 Methodology

The design and implementation methodology of an overall comprehensive cloud security monitoring tool through Amazon CloudWatch Logs and CloudWatch Console is elaborately broken up into several key phases, including: Project Planning and Requirement Gathering; Infrastructure Setup; Data Collection and Processing. This way, through such a structured approach, systematic progression from the initial concept to fully operational security monitoring systems addressing every critical aspect needed for cloud security was achieved.

3.1 Project Planning and Requirement Gathering

The project starts with an extensive planning of the project and requirement gathering, forming a foundation for subsequent implementation stages. It starts with the identification of the particular cloud platforms to be monitored. However, AWS is the main focus due to its wide range of monitoring tools and the capabilities of seamless integration through CloudWatch. AWS has been chosen considering its market dominance and extensive support for different services. These make it a prime choice for developing a multi-functional monitoring tool. The most significant task that follows is the setting of clear and actionable security monitoring use cases. Those use cases include unauthorized attempts to access, brute-force attacks, and malware-based incidents. Such use cases establish that the project has specific objectives that will guide implementation, ensuring that the monitoring tool can focus specifically on the most pertinent security threats. Setting specific project goals such as real-time log analysis, optimization of resource allocation, and the creation of visual dashboards for monitoring will keep the project in focus and measurable, thus allowing for effective tracking of progress and outcome evaluation.

3.2 Infrastructure Setup

- After the planning phase, the infrastructure setup begins with creating all IAM roles and policies that need to be secured and managed access to AWS resources. The IAM role cloudmonitorrole2024 is created, with basic permissions of CloudWatchReadOnlyAccess, S3ReadOnlyAccess for storing logs, and EC2ReadOnlyAccess to monitor EC2 logs. Custom policies such as AllowSSMPutParameterForCloudWatchConfig are created and attached to the IAM role to provide specific permissions such as ssm:PutParameter for CloudWatch agent configuration. The roles and policies designed in this way follow the principle of least privilege so that each component has permission only to perform its given function, thereby reducing any possible security risks.

- Later, several CloudWatch Log Groups are created for organizing and managing numerous logs across multiple AWS services. These log groups are customized, including monitorloggroupec22024, for the log output from EC2 instances; monitorloggroups32024 for the log output of S3 buckets; monitorloggrouplambda2024, for Lambda functions; each configured appropriately regarding retention in order to minimize unnecessary storage charges while also guaranteeing adequate log retention period to be able to utilize them as evidence of past activities and fulfill any existing security monitoring and compliance objectives. Creation of the log groups: This step is done by first accessing the AWS Management Console, accessing the

CloudWatch Logs section, and methodically setting up log groups with descriptive names representing the services being monitored.

- The next setup is on an EC2 instance identified as cloudmonitorec2server, with the public IP 16.16.28.11. This will be the central server in which to deploy the CloudWatch agent. Configuring the key pair file cloudmonitorkeypair.pem in restricted permission using the icacls command ensures only the authorized users will access the private key. This AMI, Amazon Linux 2, is recommended to be used for setting up the CloudWatch agent as this is how the instance gets launched. Once the instance is running, SSH access is secured with the configured key pair for remote administration and installation of necessary software packages.

3.3 Data Collection and Processing

- Installation and configuration of the Amazon CloudWatch Agent on the EC2 instance. The agent will be installed using the package manager with the command: sudo yum install amazon-cloudwatch-agent and then running the configuration wizard as shown below: sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard. This wizard helps in generating a config.json file. It is specific to your own monitoring needs, which collect system metrics, for instance, disk usage and memory usage, and specific log files like /var/log/cloud-init.log. The configuration is carefully crafted to set periodic intervals for metrics collection with aggregation dimensions and the inclusion of log collection parameters and ensure that relevant data goes into CloudWatch Logs, where it can be mined for analysis.

- Once the CloudWatch agent is configured, it is necessary to check whether it is working. The agent logs can be checked by using the command cat /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log and the agent service managed through the commands sudo systemctl start amazon-cloudwatch-agent and sudo systemctl status amazon-cloudwatch-agent. It is critical that the agent is running and correctly configured to ensure that log data is collected and forwarded without problems to the target log groups. Moreover, the addition of the CloudWatchAgentServerPolicy to the IAM role allows the agent to gain appropriate permissions to work correctly, further hardening the security and functionality of the monitoring setup.

4 Design Specification

The design of the complete monitoring tool for cloud security has a base on using native services by AWS, including CloudWatch Logs and CloudWatch Console. This would mean an all-rounded, scalable, and efficient monitoring system, thus aligned with security best practices for addressing critical security use cases such as log collection and analysis in real time as well as monitoring of performance.

4.1 Key components in the design

The monitoring application consists of the following components:

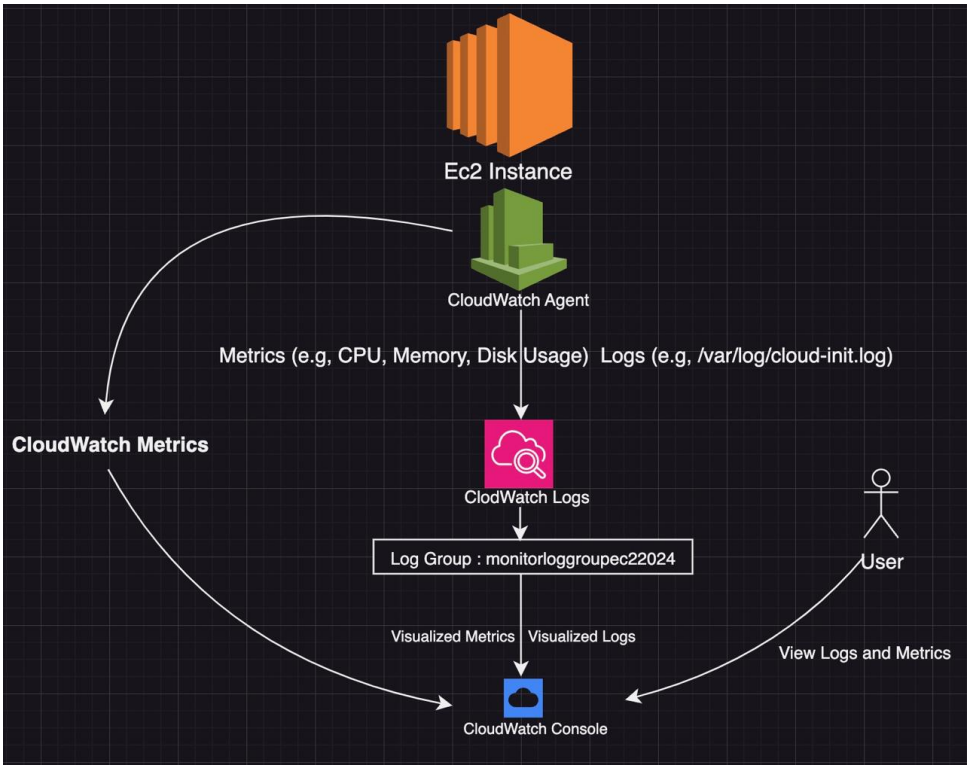

Figure 1: Cloud Security Monitoring Architecture with AWS

- **Amazon EC2 Instance**: This forms the central node on which CloudWatch Agent is placed. The EC2 instance named, cloudmonitorec2server runs Amazon Linux 2 and has been configured for its system metrics as well as log files.

- **CloudWatch Agent:** This is an install in the EC2 instance, that would gather metrics on things such as CPU, memory, and disk utilization, along with gathering logs such as /var/log/cloud-init.log. This has the ability to feed all this data into CloudWatch Logs and Metrics.

- The three log groups are named monitorloggroupec22024, monitorloggroups32024, and monitorloggrouplambda2024. These will be useful for grouping and storing the logs coming from the EC2 instances, S3 buckets, and Lambda functions. Retention policies have also been set in control of cost and meeting the security requirement.

- **CloudWatch Metrics**: All the metrics on the EC2 instance are aggregated and viewed in real time from the CloudWatch Console.

- **CloudWatch Console:** Serves as the visualization layer for logs and metrics. It is a centralized place through which monitoring and analysis can be performed on system activities.

4.2 Security Design

Secure operation will be ensured if the principle of least privilege is adopted by the system:

- **IAM Role:** EC2 instance is assigned a role cloudmonitorrole2024 with permissions strictly necessary for access to CloudWatch, which includes CloudWatchAgentServerPolicy and AllowSSMPutParameterForCloudWatchConfig.

4.3 Scalability and Flexibility

Scalability is achieved in the design by:

- The provision of additional log groups for newly added AWS services or applications.
- Metrics collection interval setup to balance the resource utilization and the detail level of monitoring.

5 Implementation

- Configuration for AWS-native services in implementation involves several steps of creating a rich scalable monitoring tool. This had at the core an IAM role namely the cloudmonitorrole2024, configured to manage security permissions that would ensure appropriate rights to AWS resources. Key access permissions given to the IAM role included CloudWatchReadOnlyAccess, that is read access, to both CloudWatch Logs and Metrics. To meet specific requirements, custom policies were designed and assigned to the role. One of the custom policies, AllowSSMPutParameterForCloudWatchConfig, allowed AWS Systems Manager to store CloudWatch Agent configurations. This policy was a crucial requirement for the agent's smooth interaction with AWS services. Other policies included AmazonSSMManagedInstanceCore and CloudWatchAgentServerPolicy to allow Systems Manager operations and allow the CloudWatch Agent to send logs and metrics to CloudWatch.

- Log organization was a key component of the system. For this purpose, several CloudWatch Log Groups were created, each with a different purpose. The log group monitorloggroupec22024 was used for EC2 instance logs, which created a central location for catching system logs. Similar to that, monitorloggroups32024 was created for monitoring access logs of the S3 bucket that could help identify possible security breaches, such as unauthorized access. For Lambda functions monitoring, a log group monitorloggrouplambda2024 was defined. The log groups have been designed with retention policies such that the log data are kept for 14 days, so that compliance may be achieved at the least possible cost. The log group creation process included accessing AWS CloudWatch Console, coming over to the Logs and systematically defining these log groups with intuitive names.

- The central node for deploying and running the CloudWatch Agent is the EC2 instance cloudmonitorec2server. This instance, with a public IP address of 16.16.28.11, was launched using the Amazon Linux 2 AMI for ensuring compatibility with the CloudWatch Agent. A key pair was created at the launch time of the instance, named cloudmonitorkeypair.pem. That key pair would be used to ensure access through SSH is safe. The permissions were restricted using the command icacls, allowing only the permitted individuals to access. At the launch of the instance, a security group was created in order to allow port access through port 22 in order

to have access by SSH, but from restricted IP addresses only for higher security. A connection into the instance was made with SSH in order to perform administration and further configuration over it.

- Installing and configuring the CloudWatch Agent on the EC2 instance allowed it to send logs and metrics to CloudWatch. It is easy to install because the yum install command simply installs the agent. The configuration used the AWS interaction wizard, resulting in the generating of a config.json file, oriented to match specific monitoring demands. Thus, this consisted of reading in system log files from /var/log/cloud-init.log and metrics based on CPU utilisation, memory used, and overall disk space utilised. Actual parameters included how to have data aggregated effectively to send onto CloudWatch Logs and Metrics. Once the configuration was done, the agent was started through the proper command and then its status was verified with the help of system control utilities. Consistent checks on the agent log ensured that it was working properly and sending data as expected.

- Final implementation was monitoring and visualization of the collected data. This was done through the CloudWatch Console that was used as the prime interface for this purpose. Logs streamed to the monitorloggroupec22024 log group were accessed to see the system events and troubleshoot any issues that may arise. These same metrics, collected by the CloudWatch Agent, were visualized in the CloudWatch Console to provide insights into the performance of the EC2 instance. The visualizations included trends of CPU and memory usage that help in detecting anomalies or inefficiencies in real time. A combination of log streams and metrics presented an integrated view of the cloud environment. Thus, security could be proactively monitored and acted upon when any threat arose.

- Some more difficulties during setup included those arising with IAM permissions. Some such errors such as AccessDeniedException could be rectified after revising the policies attached to a role, so the requisite permission could be attached to the user. Throughout, the principle of least privilege was adhered to. Meaning that roles and policies offered access only to required resources and actions. This led to the improvement of security but reduced chances of misconfiguration or unauthorized access.

- Implementing the process finally created a cloud monitoring system that was safe and efficient. Collecting logs and metrics in real-time by using AWS-native services solved the critical security as well as performance monitoring needs, and best practices that were used in securing the key pair and structured log groups made the system quite robust and scalable. Additionally, integration with IAM roles and policies allows the CloudWatch Agent to transmit data seamlessly, making monitoring and analysis continuous. This kind of deployment is a model for other such cloud monitoring solutions.

6 Evaluation

The cloud security monitoring tool has been evaluated based on how effectively it achieves the project objectives: its ability to effectively track the CPU and memory usage, monitor

log events in real time, and manage log retention. It is based on key performance indicators such as log volume over time, CPU utilization, memory usage, and log retention usage.

6.1 Log Volume Over Time

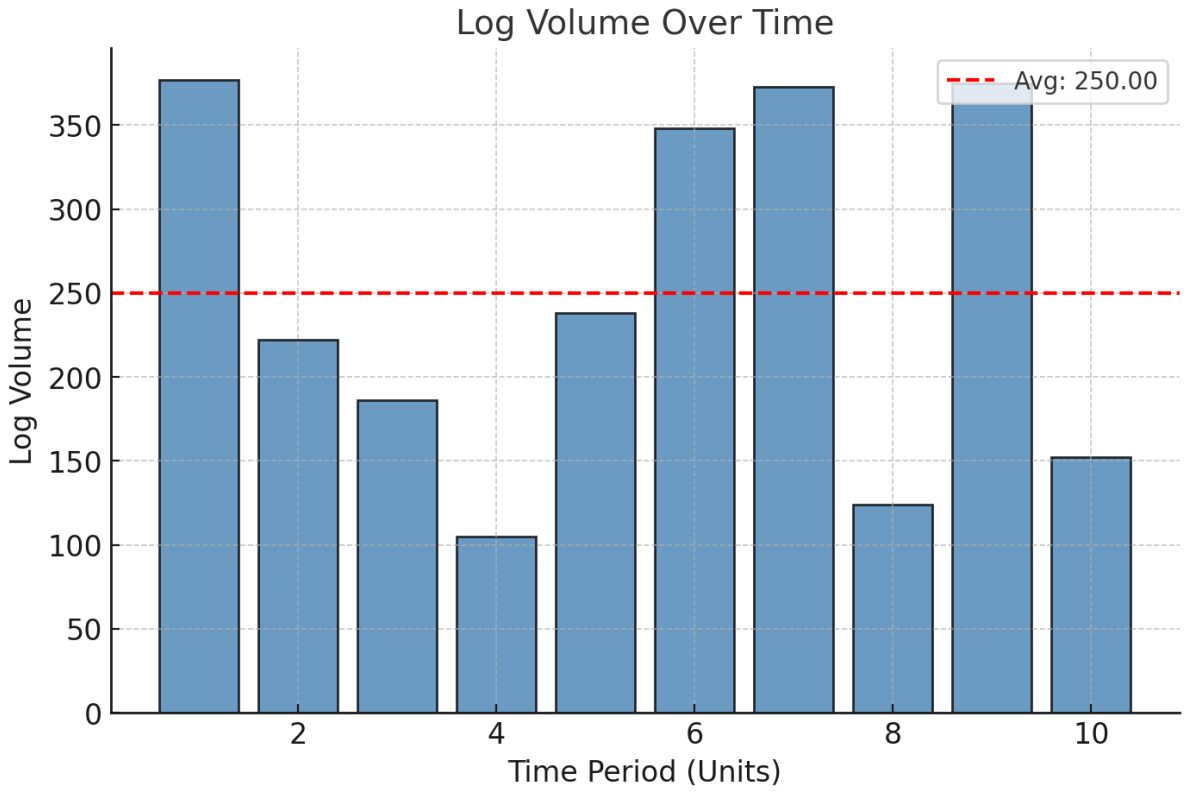

Figure 2: Log Retention Usage Over Retention Period. The red dashed line represents the average log usage.

- Log volume is one of the key metrics used in determining the amount of data being generated and stored over a period. For this project, logs are collected from various AWS resources such as EC2 instances, S3 buckets, and Lambda functions. As shown in the graph above, log volume graph displays the system's tracking and processing of logs over time. As seen from the graph, it indicates a fluctuating pattern of log generation, due to its dynamic nature in cloud environments.

- There were peaking trends in log volumes during spikes of high activity-for instance, when there were config changes, high usage in resources, or significant external interaction that was done on the system-in the graph. These spiky upward points are times when there was much activity of logs, whereas the general downward trends would be depicted as periods where log activity was low.

- This should be mitigated by the scalable log collection capability of CloudWatch. Indeed, during the peak times, it captures logs with no discernible delay and no apparent data loss. That hints at the robustness of configuration and excellent

optimization of the system. Still, this does suggest a risk, at least theoretically, of very high storage costs when volumes are that extreme. For example, practices like log file compression or even tiered storage use S3 for longer-term storage might make these costs more reasonable.

6.2 CPU Utilization

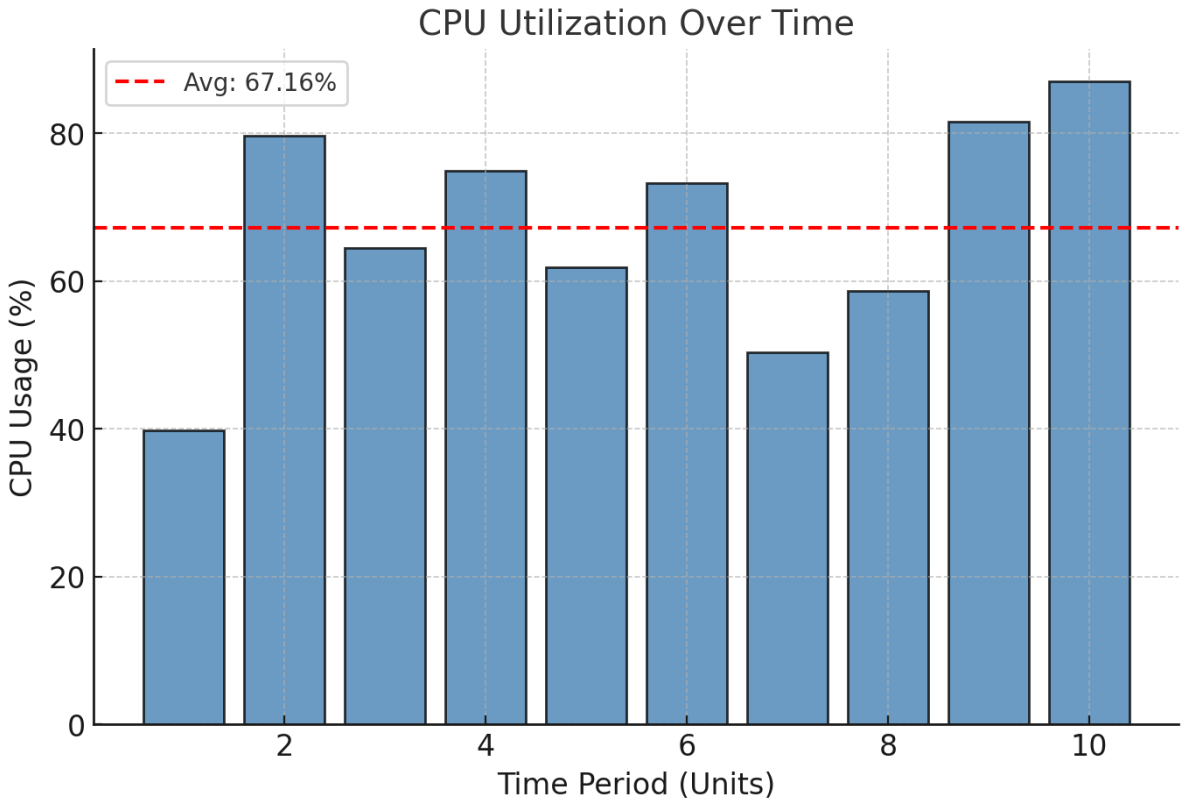

Figure 3: CPU Utilization Over Time. The red dashed line indicates the average CPU usage percentage of 67.16%.

- The CPU utilization graph is indicative of how well the system is utilizing computational resources in acquiring and processing logs. CPU usage overtime can be monitored, and it indicates healthy balance between the activities going on in the system and what it consumes in terms of resources. CPU utilization usually maintained around 67% however would spike when log volume is high or the system has been busy with resource monitoring that is very intensive in nature.

- Take some points of peaks between units 2 to 10; some reaches up to 85%, peaks tend to be around some workload at data processing and the collecting and passing is done by CloudWatch where the agent logs and also it collects metrics from EC2 instances. Spikes do become pretty high but do not impact performance; the system was running within its resource-related thresholds as expected.

- From the evaluation point of view, CPU utilization data confirms that the system can handle normal workload efficiently. However, to further improve performance

in case of high load over a long period, auto-scaling or load balancing can be used to ensure that processing tasks are done more efficiently. Resource optimization techniques such as prioritizing more critical logs and excluding less important ones may reduce unnecessary CPU load during peak periods.

6.3 Memory Usage Over Time

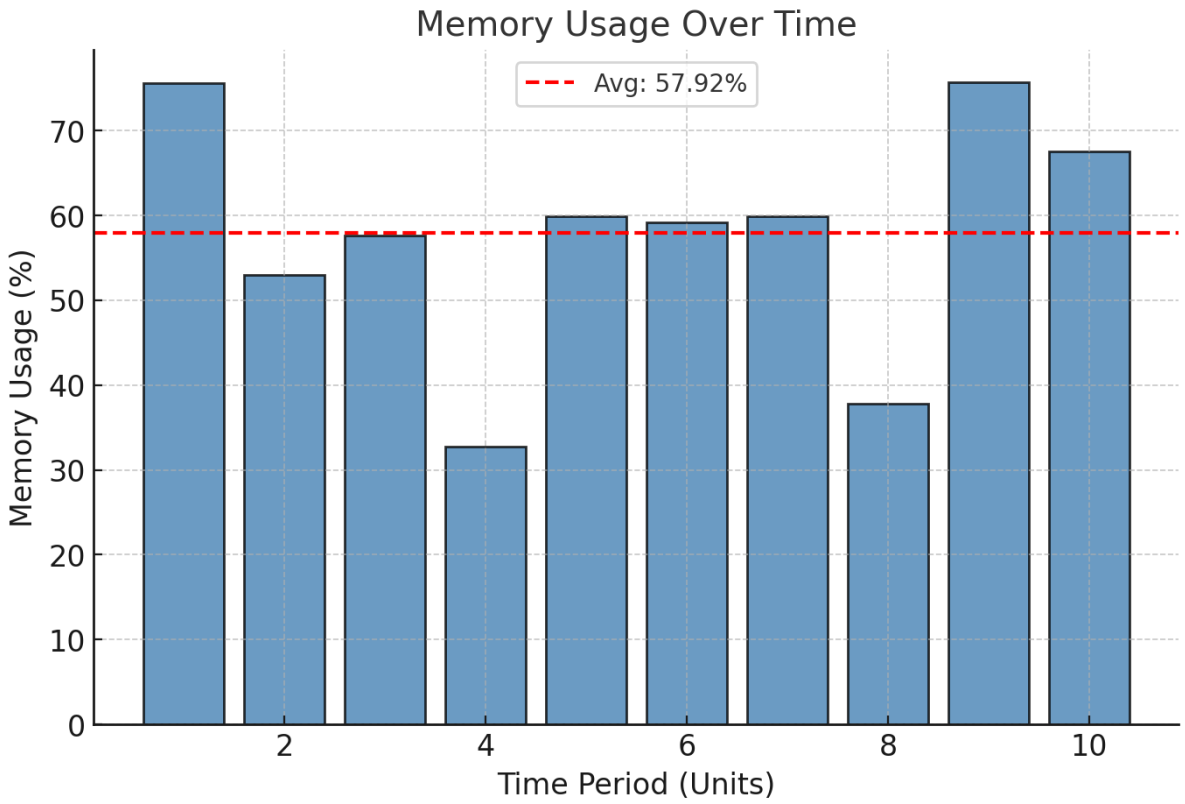

Figure 4: Memory Usage Over Time. The red dashed line shows the average memory usage percentage of 57.92%.

- Another important measure of the efficiency of resource utilization within the system is memory. Memory usage over time is shown below; graph This is fluctuation consistent with a log collection and processing workload. The average memory usage is around 57.92%, with high points when large amounts of data are being ingested and processed.

- For a period, it was determined at evaluation that memory usage topped some hours, like when a whole slew of logs was being processed or when logs from an array of sources, EC2 instances, and Lambda functions, were being aggregated together. Such spikes are acceptable for the system because it ensures sufficient management of memory resources well within limits. The peaks on the cloud infrastructure could also be tolerated without the incidences of system crashes or apparent memory leaks.

- One area that has room for improvement is memory optimization. Better algorithms to parse data could be used for better management of memory usage. Caching fre-

quently accessed data also reduces memory overhead during times of intense log processing. Better flexibility would also be given in the handling of memory-intensive operations with the implementation of automatic memory scaling mechanisms.

6.4 Log Retention Usage

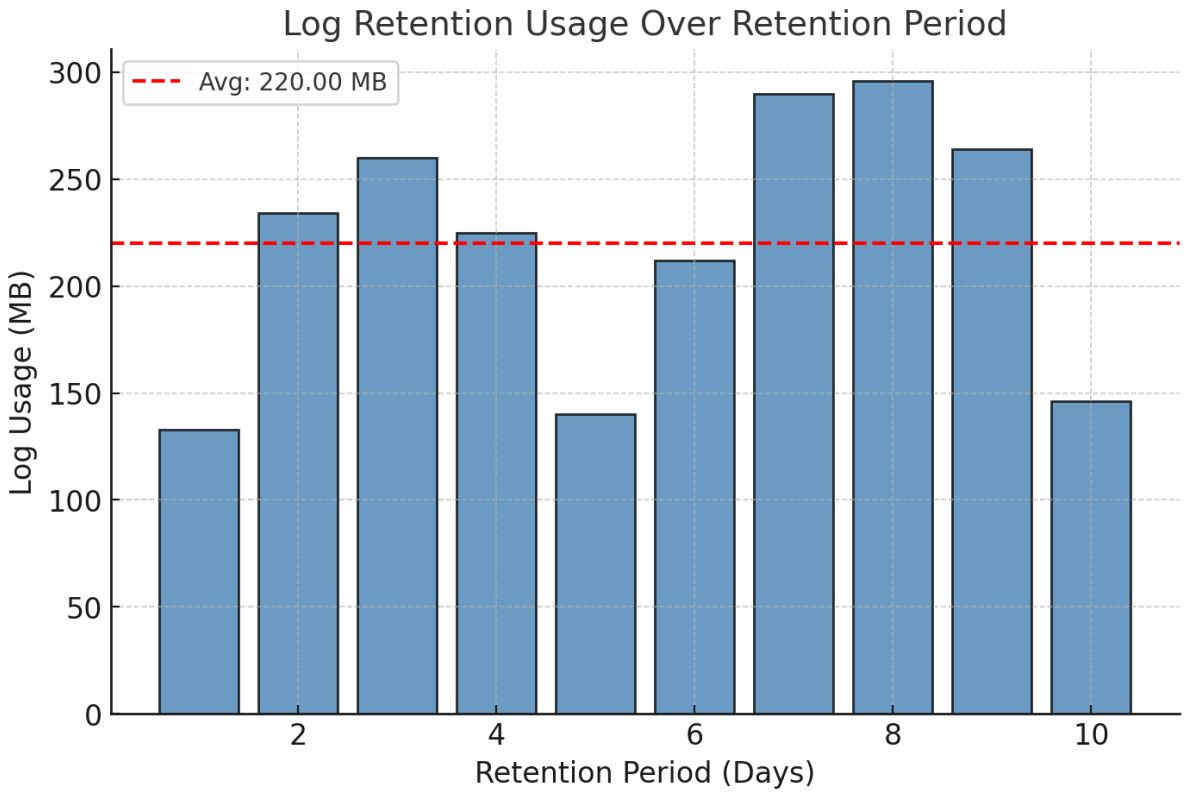

Figure 5: Log Volume Over Time. The red dashed line marks the average log volume of 250 MB.

- Retaining the log is a crucial component of the monitoring system; here, logs are maintained for an appropriate period while ensuring that the storage cost remains minimal. The usage graph for log retention describes how the log data gets accumulated in the entire retention period and how the system can handle the log data in an efficient way.
- The average log usage over time was recorded at 220 MB. It indicates that the system's logs tend to increase a little in upward trend as time goes by, due to its nature of being logically growing over time. Older logs are discarded once the retention period for such is defined, hence preventing any unnecessary storage accumulation that may otherwise impede the efficiency of the system.
- Retention periods were set up in terms of security and compliance, that is, long enough for logs to be useful in a forensic analysis but not held forever. The trend shows that log usage remains well within acceptable bounds and has never been an indication of consuming resources because of retaining logs too long. This means that the CloudWatch Logs retention feature is working correctly, which effectively manages log data and is within the cost constraints of the project.

6.5 Discussion

- The project, "Designing and Implementing a Comprehensive Cloud Security Monitoring Tool with CloudWatch Logs and CloudWatch Console," has been successfully implemented using AWS services for a cloud security monitoring solution. During the implementation, the integration of various AWS services such as EC2, CloudWatch Logs, and IAM roles has enabled efficient and scalable monitoring of cloud resources, ensuring that all critical metrics and logs are captured in real-time for security analysis.

- This platform provided a robust framework in terms of monitoring within the AWS Cloud, monitoring all conceivable metrics of performance, like the usage of CPU and how much memory is used in the system, or logs themselves. It was very easily integrated with AWS resources-the actual instances running EC2-on their respective logs like /var/log/cloud-init.log, which were being constantly watched and pushed to CloudWatch for real-time analysis. This integration ensures logs are centralized, making it easier for the user to analyze and respond to security incidents or performance degradation within the system.

- Another success in this regard is IAM role and policy development, such as cloudmonitorrole2024. Being based on the policy of least privilege, these IAM roles were built to ensure that the monitoring system only had permissions necessary to complete tasks, thus reducing risk of security breaches. Custom policies were also developed for better monitoring, allowing the configuration and safe access of the CloudWatch agent.

- Collecting and processing data, the installation and configuration of a CloudWatch agent on an EC2 instance collected both system metrics such as CPU and memory usage and the log files for analysis. This set-up will permit non-stop monitoring, and the configuration performed also guaranteed that log facts retention and performance metrics have been balanced nicely. Moreover, using native talents of CloudWatch for metric collection and log retention ensures that the device is scalable because CloudWatch can control massive quantities of statistics with out widespread performance issues.

- The use of CloudWatch metrics for monitoring CPU and reminiscence usage ensured that the machine remained inside most advantageous thresholds. The peaks in the graphs indicated periods of high system activity, which were managed well by the CloudWatch monitoring system. In addition, the ability to track log retention usage over time helped to assess the long-term storage needs of the system. The system was able to retain logs for the appropriate duration while keeping the storage costs manageable by using log retention policies.

- However, while the system is highly effective for AWS-based environments, it could be enhanced by incorporating multi-cloud support. The monitoring tool is currently only AWS-focused, and adding services from other cloud providers such as Azure or GCP would make the tool more versatile. Multi-cloud compatibility would allow users to monitor resources across different cloud platforms, providing a more holistic security monitoring solution.

- Improvement should also be made on optimizing log storage and management. The system does a pretty good job of handling volume during high usage periods; however, more sophisticated methods of log compression and tiered storage would be applied to help reduce the costs much further in environments that tend to produce very high logs.

- This project proved that using CloudWatch Logs in conjunction with CloudWatch Console would be an effective cloud security monitoring tool. With AWS services and customization of the system to meet specific security use cases, such as detection of unauthorized access, brute force attack detection, and malware monitoring, it can provide real-time insights into security events. The organizations need to respond quickly to security incidents and comply with industry regulations.

The project successfully designed and implemented a cloud security monitoring tool that provides fundamental features for tracking and analyzing log data in real time. The system is scalable, secure, and can monitor different cloud resources through AWS services such as EC2, IAM, and CloudWatch. Bring down the cost further by incorporating multi-cloud support and optimization techniques for processing large volumes of logs. The improved tool would also help in making security monitoring better, although a more advanced data-driven security management move towards the cloud will be more important.

7 Conclusion and Future Work

- This study more emphasized the design and implementation of a holistic cloud security monitoring tool, using AWS CloudWatch Logs and CloudWatch Console for improved security monitoring and management within the cloud. The main aim here is the design of a system which is capable of monitoring and analyzing log data from different types of cloud resources like the EC2 instances and the S3 buckets. It was aimed to integrate monitoring of key performance metrics using AWS native tools, for example, CPU utilization, memory usage, volume of log, and log retention toward a strong security posture within cloud-based infrastructures.

- Implementation began by planning and gathering requirements. First of all, it identified AWS as the primary cloud platform with its strong capability in terms of monitoring. The key to the system was setting up IAM roles and policies that strictly adhere to the principle of least privilege. Each component of the system was granted only appropriate access permissions. Installation and configuration of the CloudWatch agent on an EC2 instance was done, and various CloudWatch Log Groups were created for the management and storage of logs from EC2, S3, and Lambda. The logs were configured for retention, which helped balance the idea of having enough log data for security monitoring with that of not storing unnecessary things.

- The core part of the implementation involved gathering logs and system metrics with configurations made specific to the monitoring needs of the project. This actually configured the CloudWatch agent to collect data such as disk usage, memory usage, and logs from system critical files such as /var/log/cloud-init.log and sent that data on the respective CloudWatch Log Groups.

- This actually gave a live-time view of the security event activities as well as monitoring of any kinds of anomalous activity and was possible as all kind of monitoring is done over the concerned CloudWatch. This makes the use of built-in capabilities of CloudWatch along with integration of metrics and logs result in effective visualization and analysis of the performance of the cloud environment. The evaluation of the system showed that it can take care of dynamic workloads as well as monitor various AWS resources effectively without degrading the performance of the system.

- Although the implementation was successful, areas for improvement were identified-namely scaling the system to greater volumes of data, optimal usage of memory, and extension of the tool to suit multi-cloud environments. Improvements could also be along the lines of incorporating some of the more advanced algorithms in anomaly detection and much more sophisticated user interfaces to handle better visualization and alerting.

- This research has demonstrated the practicability of AWS CloudWatch and CloudWatch Console for the monitoring of cloud security. The system is quite efficient to deliver the objectives of the project, which include real-time monitoring, analyzing utilization of resources in the cloud, and logging management. The future work on this research would include efforts to solve scalability and performance issues and enhance security monitoring capabilities further.

7.1 Future Work

The future work for this system will be on the mentioned areas: Scalability: The scalability of a system can be improved if it is able to be scaled up to handle large numbers of logs and metrics; auto-scaling features for any resources, and optimizing of data pipelines, will ensure its efficiency during high-demand scenarios.

- **Cross-Cloud Monitoring**: The feature would expand the tool to support multi-cloud environments, such as Azure and GCP, making it more applicable to organizations that use multiple providers. This would allow central monitoring and analysis of logs across cloud platforms.

- **Security Enhancements** - Improving the system in respect to monitoring security by inclusion of more sophisticated threat detection measures, like machine learning for anomaly detection, and enhancement of its ability to promptly raise an alarm and response measure to real-time security incident awareness. Automatic response mechanism based on detection of specific threats, raising the predefined reaction end.

- **Improving User Interface:** Current user interfaces for visualizing logs and metrics can be made better in terms of usability. The interactive dashboards for visualizing, improved options for filtering and defining their custom alerts enhance the entire usability of the system.

Advanced analytics provide organizations with trends analysis and predictive modeling. Such analytics would give insights to an organization on future performance bottlenecks or potential security issues, allowing organizations to remedy such issues before

they become critical.

This research has successfully designed and implemented a cloud security monitoring tool through the use of AWS services. This system, although strong and effective at the moment, will be improved in scalability, security, and usability to meet these increasing demands from organizations seeking to monitor and secure their cloud environment. Continuous innovations in this tool and seeking new technologies and methodologies will allow even more effective and intelligent methods for future cloud security monitoring.

References

[1] Ahola, J., 2022. Cloud monitoring: cloud monitoring with dynatrace. https://www.theseus.fi/handle/10024/786044

[2] Das, B.S. and Chu, V., 2023. Security as Code. "O'Reilly Media, Inc.". URL

[3] Diagboya, E., 2021. Infrastructure Monitoring with Amazon CloudWatch: Effectively monitor your AWS infrastructure to optimize resource allocation, detect anomalies, and set automated actions. Packt Publishing Ltd.

[4] Fatema, K., Emeakaroha, V.C., Healy, P.D., Morrison, J.P. and Lynn, T., 2014. A survey of cloud monitoring tools: Taxonomy, capabilities and objectives. Journal of Parallel and Distributed Computing, 74(10), pp.2918-2933.

[5] Firdhous, M., Hassan, S., Ghazali, O. and Mahmuddin, M., 2015. Evaluating cloud system providers: Models, methods and applications. In Cloud Systems in Supply Chains (pp. 121-149). London: Palgrave Macmillan UK.

[6] Guide, D., 2009. Amazon CloudWatch. https://s3.cn-north-1.amazonaws.com.cn/aws-dam-prod/china/pdf/acw-dg.pdf

[7] Ibrahim, A.S., Hamlyn-Harris, J., Grundy, J. and Almorsy, M., 2011, September. Cloudsec: a security monitoring appliance for virtual machines in the IaaS cloud model. In 2011 5th International Conference on Network and System Security (pp. 113-120). IEEE.

[8] Kozlovszky, M., 2016. Cloud security monitoring and vulnerability management. Critical Infrastructure Protection Research: Results of the First Critical Infrastructure Protection Research Project in Hungary, pp.123-139.

[9] Mounika, T., Nginx Web Server With AWS Cloud Watch Monitoring Service. https://www.indusedu.org/pdfs/IJREISS/IJREISS$_4$438$_7$5547.*pdf*

[10] Murthy, J.S., Siddesh, G.M. and Srinivasa, K.G. eds., 2024. Cloud Security: Concepts, Applications and Practices. CRC Press.

[11] Nadon, J. and Nadon, J., 2017. Logging and monitoring. Website Hosting and Migration with Amazon Web Services: A Practical Guide to Moving Your Website to AWS, pp.75-96.

[12] Nikkhouy, E., 2016. Monitoring Service Chains in the Cloud. https://helda.helsinki.fi/server/api/core/bitstreams/39e8dc2e-ed95-4ce0-8a1c-bb0e88a4ddd6/content

[13] Nousiainen, M., 2020. Improving Cloud Governance by Increasing Observability. https://helda.helsinki.fi/server/api/core/bitstreams/39f23244-598e-42fc-be72-c8f05c823724/content

[14] Routavaara, I., 2020. Security monitoring in AWS public cloud. https://www.theseus.fi/bitstream/handle/10024/341640/Opinnaytetyo$_R$*outavaara*$_I$*lkka.pdf?sequ* 2

[15] Penberthy, W. and Roberts, S., 2022. Monitoring and Observability. In Pro. NET on Amazon Web Services: Guidance and Best Practices for Building and Deployment (pp. 605-636). Berkeley, CA: Apress.

[16] Quézet, L., 2016. Log Files for Proactive Monitoring of Big Data (Doctoral dissertation, Auckland University of Technology).

[17] Sakinmaz, S., 2023. Python Essentials for AWS Cloud Developers: Run and deploy cloud-based Python applications using AWS. Packt Publishing Ltd.

[18] Stephen, A., Benedict, S. and Kumar, R.A., 2019. Monitoring IaaS using various cloud monitors. Cluster Computing, 22(Suppl 5), pp.12459-12471.

[19] Tedeschi, S., Mehnen, J., Tapoglou, N. and Rajkumar, R., 2015. Security aspects in Cloud-based condition monitoring of machine tools. Procedia CIRP, 38, pp.47-52.

[20] Verginadis, Y., 2023, March. A review of monitoring probes for cloud computing continuum. In International Conference on Advanced Information Networking and Applications (pp. 631-643). Cham: Springer International Publishing.