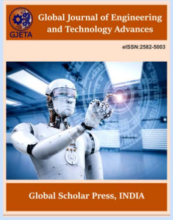
Global Journal of Engineering and Technology Advances

eISSN: 2582-5003

Cross Ref DOI: 10.30574/gjeta

Journal homepage: https://gjeta.com/



(Research Article)

# Automated API framework tools for evaluating cloud resources (IAM, S3, KMS) for compliance with ISO 27001 case study AWS

Trudy-Ann Campbell [1,*], Samson Eromonsei [1] and Olusegun Afolabi [2]

[1] *School of Engineering Prairie View A&M University Prairie View, Texas USA.*

[2] *Department of Information Systems and Business Analysis, Aston Business School, Aston University, Birmingham, UK.*

Global Journal of Engineering and Technology Advances, 2024, 20(01), 131–149

Publication history: Received on 09 June 2024; revised on 17 July 2024; accepted on 20 July 2024

Article DOI: https://doi.org/10.30574/gjeta.2024.20.1.0126

## Abstract

CLOUD— computing's advancements has provided scalability and adaptability but has also given rise to data security concerns. ISO 27001 is vital for cloud information security, yet compliance in dynamic settings poses challenges. Automated API framework tools automate ISO 27001 compliance checks for IAM, S3, and KMS services in AWS, boosting efficiency and minimizing errors. This study investigates the effectiveness of these frameworks, focusing on AWS environments. It explores advantages, difficulties, and practical considerations of automation in cloud compliance. Insights aim to enhance understanding of how automation reinforces security and regulatory adherence. Previous studies highlight the need for adaptable monitoring solutions in cloud setups. Recent research demonstrates the potential of programming languages like Python to streamline compliance processes effectively. This study contributes by examining the efficiency of automated compliance frameworks in AWS, offering perspectives on their practical application in cloud settings.

**Keywords:** Cloud Computing; ISO 27001 Compliance; Automated Compliance Frameworks; AWS Services (IAM; S3; KMS); Security Management Systems; Regulatory Compliance; Data Security; Automation in Cloud Computing; Information Security; Cloud Security

## 1. Introduction

### 1.1. Background of Cloud Resource Management

Cloud resource management is an essential aspect of cloud computing, involving the efficient allocation, provisioning, and monitoring of resources to meet the diverse needs of users and applications. The primary components of resource management include resource provisioning, resource scheduling, and resource monitoring.

Resource provisioning is the process of identifying and allocating the necessary resources to execute a specific workload based on Quality of Service (QoS) requirements defined by cloud consumers. This stage involves the discovery and selection of resources that match the workload's needs, ensuring that the appropriate resources are available for execution (Singh & Chana, 2017). The provisioning of resources is a critical step in maintaining performance and cost efficiency in cloud environments.

Resource scheduling follows provisioning and involves mapping, allocating, and executing workloads based on the selected resources. The goal is to optimize the use of resources while adhering to the Service Level Agreements (SLAs) defined by users. Effective scheduling ensures that workloads are processed efficiently, minimizing costs and execution times (Jennings & Stadler, 2015). Resource scheduling must account for various factors such as resource availability, workload priorities, and performance requirements.

[*] Corresponding author: Trudy-Ann Campbell

Resource monitoring is an ongoing process that ensures resources are being used effectively and efficiently. It involves tracking the status of workloads, verifying that the allocated resources meet the required QoS, and adjusting resource allocations as necessary to maintain performance. Monitoring helps in detecting deviations from expected performance and allows for proactive management to address any issues that arise (Manvi & Shyam, 2014). This phase is crucial for maintaining the reliability and availability of cloud services.

Overall, cloud resource management aims to balance the needs of cloud providers and consumers by efficiently utilizing resources, ensuring performance, and minimizing costs. The dynamic nature of cloud environments requires robust and adaptive management strategies to handle the varying demands of applications and users effectively.

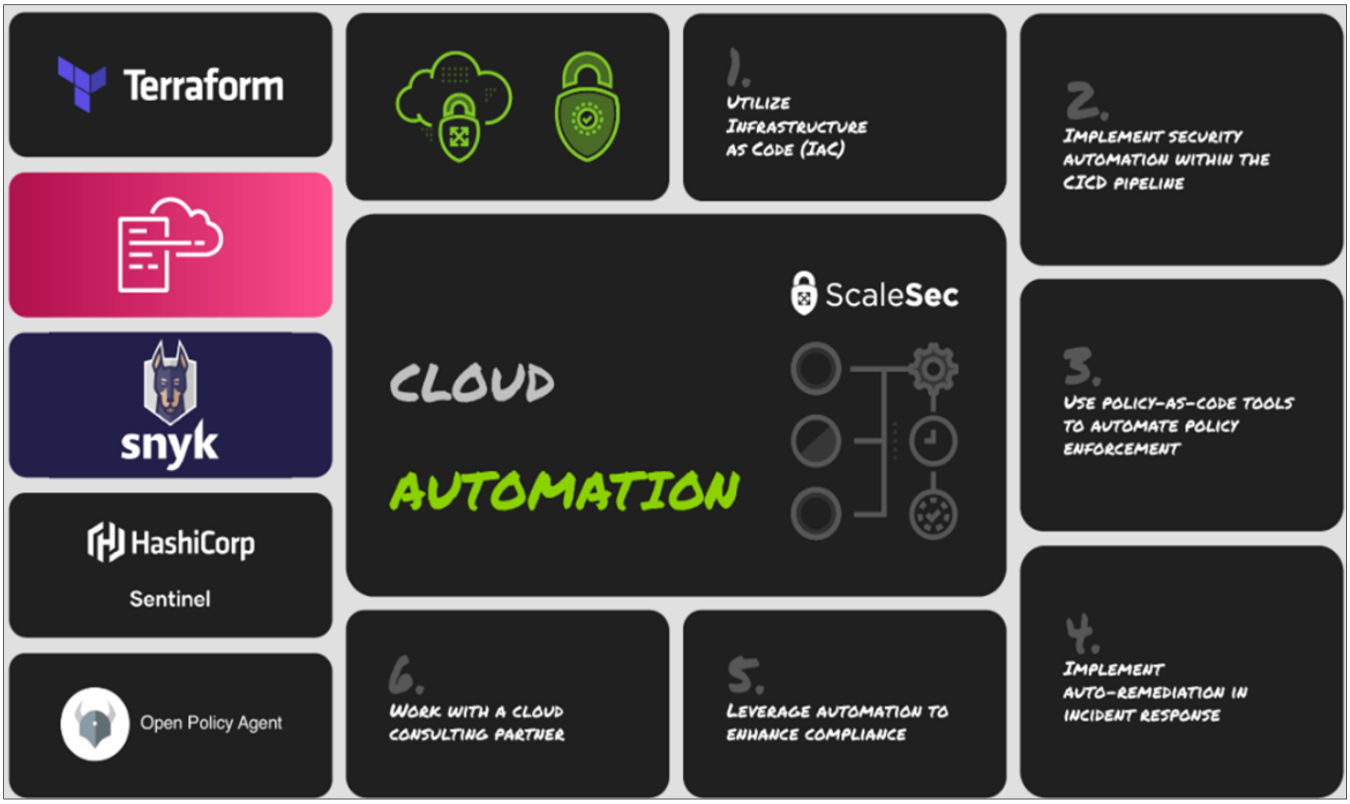

Terraform
1. Utilize Infrastructure as Code (IaC)
2. Implement security automation within the CICD pipeline
ScaleSec
Cloud Automation
3. Use policy-as-code tools to automate policy enforcement
snyk
HashiCorp Sentinel
4. Implement auto-remediation in incident response
Open Policy Agent
6. Work with a cloud consulting partner
5. Leverage automation to enhance compliance

**Figure 1** Strategies for Enhancing Cloud Security and Compliance through Automation (ScaleSec., 2024)

Figure 1 illustrates a comprehensive approach to enhancing cloud security and compliance through automation. It emphasizes six key strategies: utilizing Infrastructure as Code (IaC) to manage and provision cloud resources through code; implementing security automation within the CI/CD pipeline to integrate continuous security checks; using policy-as-code tools like Open Policy Agent (OPA) and HashiCorp Sentinel to automate policy enforcement; and implementing auto-remediation in incident response to quickly mitigate threats. Additionally, it suggests leveraging automation to enhance compliance by conducting automated compliance checks and reports, and working with a cloud consulting partner to gain expert insights and strategies. The diagram also highlights tools such as Terraform for IaC, Snyk for security automation, and various policy enforcement tools, showcasing a holistic approach to cloud security automation.

### 1.2. Importance of Compliance with ISO 27001

Compliance with ISO 27001 is crucial for organizations, especially those leveraging cloud computing, as it provides a comprehensive framework for managing information security. ISO 27001 helps organizations protect their data, ensuring confidentiality, integrity, and availability of information across all platforms, including cloud environments (Cloudlytics, 2022).

ISO 27001 compliance offers several significant benefits. Firstly, it helps organizations manage risks associated with information security by identifying, assessing, and addressing potential vulnerabilities (ISO Council, 2022). This risk-based approach is essential in the dynamic landscape of cloud computing, where threats are continuously evolving. By adhering to ISO 27001 standards, businesses can mitigate risks and enhance their resilience against cyberattacks.

Secondly, ISO 27001 certification can significantly enhance an organization's credibility and trustworthiness. It demonstrates a commitment to maintaining high standards of information security, which is crucial for building trust with clients, partners, and regulatory bodies. Organizations that are ISO 27001 certified are better positioned to assure stakeholders that they are proactively managing security risks associated with cloud services (Sysdig, 2022).

Lastly, compliance with ISO 27001 facilitates adherence to various regulatory requirements. Many industries have strict regulations governing the storage, processing, and handling of sensitive data. ISO 27001 provides a robust framework that aligns with these regulatory standards, helping organizations ensure that their cloud-based operations remain compliant with applicable laws and regulations (ISO Council, 2022). This not only helps in avoiding potential fines and penalties but also ensures the integrity and security of sensitive information.

In summary, ISO 27001 compliance is essential for organizations utilizing cloud services as it enhances risk management, credibility, and regulatory compliance. Implementing an Information Security Management System (ISMS) in accordance with ISO 27001 ensures that organizations can effectively protect their data and maintain a secure cloud environment.

## 1.3. Overview of AWS Services: IAM, S3, and KMS

Amazon Web Services (AWS) offers a comprehensive suite of services to help organizations manage their cloud resources effectively. Three critical services among these are Identity and Access Management (IAM), Simple Storage Service (S3), and Key Management Service (KMS), each serving distinct but complementary roles in cloud infrastructure management.

### *1.3.1. AWS Identity and Access Management (IAM)*

IAM is a vital service that helps you securely control access to AWS services and resources. It allows you to create and manage AWS users and groups and use permissions to allow or deny their access to AWS resources. With IAM, you can manage users' credentials, such as passwords and multi-factor authentication, and you can set permissions for individual users or groups to access specific AWS services (AWS, 2024).

### *1.3.2. Amazon Simple Storage Service (S3)*

Amazon S3 is a scalable storage service designed to store and retrieve any amount of data from anywhere on the web. It provides a simple web service interface that you can use to store and retrieve data, making it ideal for a wide range of use cases, including backup and restore, archiving, data lakes, and cloud-native applications. S3 offers various storage classes, allowing users to optimize costs by storing data in different classes based on accessibility and retrieval needs (AWS, 2024).

### *1.3.3. AWS Key Management Service (KMS)*

AWS KMS is a managed service that allows you to create and control the encryption keys used to encrypt your data. It integrates with several other AWS services to provide a consistent way to protect data across AWS environments. KMS supports both symmetric and asymmetric encryption keys and offers features like automatic key rotation and centralized key management. By using KMS, you can gain control over who can use your encryption keys and access your encrypted data, ensuring robust security for sensitive information (AWS, 2024).

### *1.3.4. Integration and Synergies*

These services work together to provide a secure and efficient cloud environment. IAM policies can be used to control access to S3 buckets and objects, ensuring that only authorized users can access or modify data. KMS can be integrated with S3 to provide server-side encryption for data at rest, ensuring that sensitive information is protected even if unauthorized access is attempted (AWS, 2024; AWS, 2024).

IAM, S3, and KMS are fundamental components of AWS that provide essential functionalities for access management, data storage, and encryption, respectively. Their integration allows for a secure and scalable cloud infrastructure that meets the diverse needs of modern enterprises.

## 1.4. Need for Automated API Framework Tools

In today's rapidly evolving cloud environment, the need for automated API framework tools for evaluating cloud resources for compliance with standards such as ISO 27001 is becoming increasingly critical. These tools provide several key benefits, enhancing security, efficiency, and compliance.

#### *1.4.1. Continuous Compliance Monitoring*

Automated API tools facilitate continuous compliance monitoring, which is essential in the dynamic cloud landscape. These tools enable organizations to constantly evaluate their cloud operations, ensuring that they remain compliant with ISO 27001 and other regulatory standards. This continuous monitoring is vital as it helps in promptly identifying and mitigating risks associated with non-compliance, thereby maintaining a robust security posture (Snyk, 2024).

#### *1.4.2. Efficiency and Scalability*

Manual compliance processes are time-consuming and prone to human error. Automated tools, on the other hand, streamline these processes, making them more efficient and scalable. They automate the collection and analysis of evidence, perform continuous checks, and generate real-time compliance reports. This not only saves time but also ensures that compliance checks are comprehensive and consistent across all cloud resources (Datamation, 2024).

#### *1.4.3. Integration with Development Processes*

Modern cloud compliance tools are designed to integrate seamlessly with existing development and operational processes, such as CI/CD pipelines and DevOps practices. This integration ensures that compliance is built into the development lifecycle from the beginning, reducing the risk of security breaches and non-compliance. By embedding compliance into daily operations, organizations can better manage their security and compliance posture in real time (Palo Alto Networks, 2024).

#### *1.4.4. Enhanced Security Posture*

Automated API framework tools enhance the security posture of organizations by providing real-time insights into potential vulnerabilities and misconfigurations. These tools can automatically detect and remediate security issues, ensuring that cloud resources remain secure and compliant. This proactive approach to security management helps organizations stay ahead of potential threats and maintain compliance with standards like ISO 27001 (Hyperproof, 2024).

The implementation of automated API framework tools for evaluating cloud resources for compliance is essential for maintaining robust security and compliance in today's cloud environments. These tools provide continuous monitoring, improve efficiency, integrate with development processes, and enhance overall security posture, ensuring that organizations can effectively manage their cloud compliance requirements.

### 1.5. Objectives and Scope of the Case Study

The primary objective of this case study is to explore and evaluate the effectiveness of automated API framework tools in assessing and ensuring compliance of AWS cloud resources—specifically IAM, S3, and KMS—with ISO 27001 standards. This evaluation will highlight the benefits, challenges, and potential improvements in the compliance processes when leveraging automated tools. By focusing on AWS services, the study aims to provide detailed insights into how these tools can be integrated into existing cloud environments to enhance security and compliance management.

The scope of the study includes a comprehensive analysis of the functionalities of automated API tools, their integration capabilities with AWS services, and their impact on maintaining ISO 27001 compliance. It will cover the implementation processes, the types of compliance checks performed, and the overall effectiveness in detecting and mitigating compliance issues. Additionally, the study will examine real-world scenarios and use cases to illustrate the practical applications and benefits of these tools in a cloud-based infrastructure. The findings from this case study will be valuable for organizations looking to improve their cloud security posture and streamline their compliance efforts using automated solutions.

### 1.6. Related Work

The rise of cloud computing has led to a review of compliance frameworks with a focus on meeting ISO 27001 standards. Papanikolaou and colleagues introduced a toolkit aimed at automating compliance processes in cloud services emphasizing the need to streamline procedures to meet demands efficiently [1]. This push for efficiency resonates with the ideas presented by other research that suggest a monitoring framework tailored for cloud setups [2]. Their research highlights the nature of cloud environments and stresses the importance of adaptable monitoring solutions in maintaining security standards compliance.

Recognizing the complexities involved in monitoring cloud frameworks, other researchers delve into the state and challenges, shedding light on overseeing cloud resources for compliance purposes [3]. Naser's study emphasizes the requirement for monitoring approaches to adapt to the changing landscape of offerings. Ristov further investigate the performance aspects of compliance solutions in elastic cloud settings providing insights into the scalability and effectiveness of compliance frameworks within clouds [4]. Their analysis emphasize how aligning compliance strategies with dynamic cloud environments is crucial.

As businesses navigate through ISO 27001 compliance challenges in cloud computing, Beckers and team, in a study introduced a method that supports context setting and asset recognition through patterns [5]. This systematic approach aims to tackle compliance issues by outlining methodologies, for meeting ISO 27001 standards in cloud environments. Expanding on this concept another study in 2023 introduced a Cloud Compliance Framework that utilizes Python to demonstrate how automation can streamline compliance processes effectively [6]. Their findings show the potential of using programming languages to create tailored compliance solutions focusing on efficiency and adaptability.

Regarding access control policies a study from 2017 emphasizes the need for a verification approach to ensure adherence to security standards rigorously [7]. Their work contributes to enhancing access control measures in cloud settings to meet ISO 27001 criteria. Similarly, another study from 2014 suggests utilizing pattern-based risk analysis for cloud systems providing a method for identifying and addressing security risks systematically [8]. This research highlights the significance of risk analysis in achieving and upholding ISO 27001 compliance requirements aligning with the goal of establishing security frameworks in cloud setups.

Lastly insights shared by researchers, in 2014 explore how cloud services can align with established security standards through studies. Their discoveries highlight the importance of cloud service providers following ISO 27001 guidelines to maintain security measures [9]. Together these studies emphasize the significance of utilizing automated API framework tools to assess cloud resources for compliance, with ISO 27001 standards. This study seeks to expand on this groundwork by examining the efficiency of tools with a focus on AWS as a case example. Through examination this investigation aims to offer perspectives on the practical application of automated compliance frameworks, in cloud settings.

### 1.7. Research Question

How does automated API framework tools enhance the effective evaluation of cloud resources, such as IAM, S3 and KMS, for compliance with ISO 27001 standards, as demonstrated through a case study of AWS?

### 1.8. Organization of the Work

This research is structured into five main sections to comprehensively address the topic of using automated API framework tools to evaluate AWS cloud resources for ISO 27001 compliance. The first section introduces the background, importance, and objectives of the study, providing a foundation for understanding the necessity of compliance and the role of automated tools. It also outlines the specific AWS services under evaluation—IAM, S3, and KMS—and sets the stage for the detailed analysis that follows.

The subsequent sections delve into the core of the research. The second section reviews various automated API framework tools, discussing their functionalities, integration capabilities, and relevance to cloud compliance. The third section focuses on evaluating AWS IAM, S3, and KMS for ISO 27001 compliance, illustrating the methodologies and tools used. The fourth section presents a case study on AWS implementation, detailing the practical application, results, and lessons learned. Finally, the fifth section concludes the research with a summary of findings, recommendations for practitioners, and future research directions. This organization ensures a logical flow from theoretical foundations to practical applications, providing a thorough examination of the subject matter.

## 2. Automated API framework tools

### 2.1. Definition and Purpose of API Framework Tools

API framework tools are essential for managing the complexity and ensuring the security of modern cloud environments. These tools provide a structured approach to developing, deploying, and maintaining APIs, which are crucial for facilitating communication between different software components and services. The primary purpose of API framework tools is to standardize the creation and management of APIs, ensuring they adhere to organizational policies and compliance requirements (Postman, 2024).

One of the core benefits of API framework tools is that they enforce governance and compliance throughout the API lifecycle. This includes defining policies for API development, deployment, and retirement, as well as ensuring these policies are followed consistently. By automating these processes, API framework tools help organizations maintain a high level of security and compliance, reducing the risk of data breaches and other security incidents (RapidAPI, 2024).

Moreover, API framework tools facilitate continuous monitoring and management of APIs, which is critical in dynamic cloud environments. They provide real-time insights into API performance, security, and compliance, enabling organizations to quickly identify and remediate issues. This capability is especially important for ensuring that APIs remain compliant with standards such as ISO 27001, which requires ongoing monitoring and improvement of information security practices (Cloudflare, 2024).

API framework tools are vital for ensuring the efficient, secure, and compliant operation of APIs in cloud environments. They provide the necessary governance, automation, and monitoring capabilities to manage APIs effectively, thereby supporting the overall security and compliance objectives of the organization.

## 2.2. Key Features of Effective API Tools

Effective API tools are critical for managing and securing APIs in cloud environments, ensuring compliance with standards like ISO 27001. These tools possess several key features that enhance their functionality and reliability.

### *2.2.1. API Security*

One of the most important features of API tools is robust security. This includes capabilities such as authentication, authorization, and encryption to protect APIs from unauthorized access and malicious activities. Effective API tools provide multiple layers of security to safeguard data in transit and at rest. These tools often integrate with security frameworks and standards, enabling organizations to maintain compliance with regulatory requirements (Stoplight, 2024).

### *2.2.2. API Monitoring and Analytics*

Monitoring and analytics are essential features that help organizations track API usage, performance, and potential issues. Effective API tools provide real-time monitoring and detailed analytics, allowing organizations to detect anomalies, understand usage patterns, and optimize API performance. This continuous visibility helps in maintaining the security and efficiency of APIs, ensuring they meet both operational and compliance standards (Solo.io, 2024).

### *2.2.3. API Governance and Policy Management*

API governance involves the creation and enforcement of policies across the API lifecycle. Effective API tools enable organizations to define, implement, and monitor governance policies that ensure APIs are developed, deployed, and maintained according to best practices and compliance requirements. These tools support policy automation, reducing the risk of human error and ensuring consistent application of security and operational standards across all APIs (Postman, 2024).

### *2.2.4. API Versioning and Lifecycle Management*

Managing different versions of an API is crucial for maintaining backward compatibility and supporting ongoing development. Effective API tools provide features for versioning APIs and managing their lifecycle from development through retirement. This ensures that APIs can evolve without disrupting existing services or compromising security and compliance (Microsoft, 2024).

### *2.2.5. Scalability and Performance Optimization*

Scalability is a key feature that allows API tools to handle increasing numbers of requests and data volumes without compromising performance. Effective API tools include mechanisms for load balancing, rate limiting, and performance tuning, ensuring that APIs can scale efficiently to meet growing demands. These tools also provide insights and optimizations to enhance API performance, crucial for maintaining service reliability and user satisfaction (RapidAPI, 2024).

Effective API tools combine robust security, comprehensive monitoring, strong governance, versioning capabilities, and scalability to ensure that APIs are secure, compliant, and efficient. These features are essential for organizations to manage their APIs effectively in cloud environments, supporting both operational excellence and regulatory compliance.

### 2.3. Methodology for Compliance Evaluation

Evaluating compliance in cloud computing involves a structured approach to ensure that cloud services meet regulatory standards and security requirements. This process is crucial for maintaining the integrity, confidentiality, and availability of data within cloud environments. The methodology typically includes several key steps.

Firstly, a comprehensive risk assessment is conducted to identify potential threats and vulnerabilities within the cloud infrastructure. This involves analyzing the security posture of cloud services and identifying areas that may be susceptible to breaches or non-compliance. The risk assessment helps in prioritizing the security controls that need to be implemented to mitigate identified risks (Kuyoro et al., 2011).

Next, the implementation of security controls is tailored to address the specific compliance requirements of standards such as ISO 27001. These controls include measures for access control, data encryption, network security, and continuous monitoring. The goal is to establish a robust security framework that not only meets regulatory standards but also enhances the overall security of the cloud environment (NIST, 2021).

Continuous monitoring and auditing are essential components of the compliance evaluation process. These activities involve regularly checking the cloud environment for compliance with established security policies and standards. Automated tools are often used to perform these checks, providing real-time insights into the security status and helping to detect any deviations from compliance requirements promptly (Chiregi & Navimipour, 2018).

Finally, documentation and reporting play a critical role in the compliance evaluation process. Detailed records of risk assessments, security control implementations, monitoring activities, and audit results must be maintained. These documents are crucial for demonstrating compliance to regulatory bodies and for conducting internal reviews to ensure ongoing adherence to compliance standards (Mitchell, 2015).

The methodology for compliance evaluation in cloud computing involves a systematic approach that includes risk assessment, implementation of security controls, continuous monitoring, and thorough documentation. This structured approach ensures that cloud services remain secure and compliant with regulatory standards.

### 2.4. Implementation of API Framework Tools for Compliance Checks

Implementing API framework tools for compliance checks involves several critical steps to ensure that cloud resources meet regulatory and security standards. This process integrates automated tools and methodologies to streamline compliance management, enhancing both efficiency and security.

#### *2.4.1. Initial Setup and Configuration*

The first step in implementing API framework tools is the initial setup and configuration. This involves defining the scope of compliance checks and configuring the tools to align with specific regulatory requirements such as ISO 27001. Tools like Postman and SoapUI provide extensive features for setting up automated tests and compliance checks (Postman, 2024; APITier, 2024). Configuring these tools involves setting parameters for authentication, data protection, and access control to ensure they meet the compliance criteria.

#### *2.4.2. Automated Compliance Checks*

Automated compliance checks are at the core of using API framework tools. These tools perform continuous monitoring and validation of API requests and responses against predefined compliance rules. For instance, ensuring that all API endpoints require proper authentication and that sensitive data is encrypted during transmission (Microsoft, 2024). Automation helps in detecting and addressing compliance issues in real-time, thereby reducing the risk of non-compliance and enhancing the security posture of the cloud environment.

#### *2.4.3. Integration with Development Processes*

Effective implementation of API framework tools also requires integration with existing development processes. This means embedding compliance checks into CI/CD pipelines to ensure that every code deployment is compliant with regulatory standards. Tools like Katalon and Rest-Assured can be integrated into the development workflow to automate testing and compliance validation (APITier, 2024). This integration ensures that compliance is maintained throughout the development lifecycle, from initial coding to production deployment.

#### *2.4.4. Regular Audits and Reporting*

Regular audits and reporting are essential components of the compliance framework. API tools generate detailed reports on compliance status, highlighting any deviations from the compliance standards. These reports are crucial for internal reviews and external audits, providing a clear record of compliance efforts and outcomes (Sprinto, 2024). Regular audits help in maintaining a continuous compliance posture, enabling organizations to quickly identify and remediate any compliance gaps.

Implementing API framework tools for compliance checks involves configuring the tools to meet specific regulatory requirements, automating compliance checks, integrating these tools into development processes, and conducting regular audits and reporting. These steps ensure that cloud resources remain secure and compliant, thereby protecting sensitive data and maintaining regulatory compliance.

### 2.5. Examples of Compliance Checks for IAM, S3, and KMS

Compliance checks for AWS services like IAM, S3, and KMS are essential to ensure that the cloud environment adheres to security and regulatory standards. Implementing these checks involves using various tools and configurations to verify that the services are secure and meet compliance requirements.

#### *2.5.1. IAM (Identity and Access Management) Compliance Checks*

For IAM, compliance checks focus on verifying that access policies are appropriately configured to enforce the principle of least privilege. This includes ensuring that IAM policies do not grant excessive permissions and that roles and users have only the necessary access rights. Tools like AWS IAM Access Analyzer can be used to continuously monitor and analyze the permissions of IAM roles, users, and policies to detect any overly permissive configurations or unintended access (AWS, 2024).

#### *2.5.2. S3 (Simple Storage Service) Compliance Checks*

S3 compliance checks primarily involve ensuring that data stored in S3 buckets is encrypted and that access to these buckets is properly controlled. This includes enabling server-side encryption (SSE) using either SSE-S3 or SSE-KMS (AWS KMS) for all objects in the bucket. Additionally, bucket policies should enforce secure transport (HTTPS) for data in transit and block public access to sensitive data. Regular audits can be performed using tools like AWS Config to check that all S3 buckets comply with these security policies (Blink, 2024; Databricks, 2024).

#### *2.5.3. KMS (Key Management Service) Compliance Checks*

For KMS, compliance checks ensure that encryption keys are properly managed and protected. This involves verifying that keys are rotated regularly, access to keys is restricted through strict IAM policies, and keys are not scheduled for deletion without proper authorization. AWS Config rules and AWS Security Hub can be used to monitor the status of KMS keys and ensure they comply with organizational policies and regulatory requirements (AWS Security Blog, 2024).

Compliance checks for IAM, S3, and KMS involve a combination of policy enforcement, encryption verification, and continuous monitoring. These checks help maintain the security and compliance of AWS cloud environments by ensuring that access controls are correctly configured, data is encrypted, and keys are managed securely.

## 3. Method

### 3.1. Setting up the Environment and configuration

Before diving into the process of compliance verification it's important to set up the environment. This step ensures that all necessary software and configurations are, in place creating a base for the steps. To run this framework, you will need the following software and libraries:

- Python 3.x: Ensure Python 3 is installed on your machine. - Boto3: AWS SDK for Python, used for direct interaction with AWS services.
- Dotenv: For loading environment variables from a .env file.
- Install the required packages using pip: bash and pip - install boto3 python-dotenv.
- sklearn Decision Tree Classifier

To configure your environment, set up the AWS credentials by creating a .env file in the root directory of your project. Add the following keys with your AWS credentials and region:

- Secret_access_key=YOUR_AWS_SECRET, - Access_key=YOUR_AWS_ACCESS_KEY, - Region=YOUR_AWS_REGION.

### 3.2. Data Collection, Application Design

In the stage of this process of data gathering metadata were extracted from the AWS resources for IAM, S3, KMS and CloudTrail. This fundamental step establishes the foundation for conducting compliance assessments and gaining an understanding of the landscape before moving forward.

A python classes and methods representing AWS resources were developed to implement data extraction see details the GitHub repository Link. Once the data is collected it's time to conceptualize the system to compare data from AWS with ISO27001 recommendation of best practices.

To achieve this comparing, we ISO27001 was distilled from its text context into actionable tabular as shown below see Table 6 in the appendix.

Here's a step-by-step approach on how you might handle this:

- Distill ISO27001 Framework to Actionable Data:

We Extracted key requirements from ISO27001 text and organize them into a tabular format. Each row might represent a specific security control, and columns might include details like control ID, description, required actions, and AWS resource mapping.

- Tabular Data to Machine Learning Input: Clean and preprocess this tabular data to be suitable for machine learning. This involved encoding categorical variables, handling missing data, and scaling or normalizing data.

Labels were defined in the range 100 to compliance, 50 for partial compliance and 0 for non-compliance for your data.

- Developing the ML Model: Decision Tree model was considered due to its interpretability and ease of use. compliance status based on AWS resource configurations.

The model was trained using a portion of your labeled data and validate its performance using a separate validation set.

- Integrating ML Predictions with Compliance Checking: We implemented a system where AWS resource metadata is automatically extracted and fed into the ML model to predict compliance.

The predictions can guide compliance audits, where resources predicted as non-compliant are prioritized for review.

- Framework System Design

The framework design was completed using object-oriented programming principles. Each compliance check (e.g., S3 bucket encryption, IAM policies) could be encapsulated as a method within a broader ComplianceManager class. Attributes of the class included the configurations for AWS connection, compliance feature tracking, and methods for each type of compliance check.

- Automation and Continuous Monitoring: This can be automated to extract AWS metadata and compliance checking using scheduled tasks or triggers based on AWS resource changes.

Continuously update the ML model with new data and compliance outcomes to improve its accuracy and relevance.

A "session Authenticator" attribute in the class object to establish a session object for connecting to AWS with the credentials.

The "container" attribute serves as a dictionary to store results from AWS resources metadata.

#### *3.2.1. Modules in the Class objects*

Compliance Manager Class is a generic class with two type variables, T and U. It includes:

compliance feature: A class variable dictionary tracking compliance status of various AWS security features.

Credentials and AWS Session

secrets, access_key, and region: Loaded from environment variables to establish a session with AWS. A session (session_connection) is established if the credentials are present, otherwise an exception is raised.

#### *3.2.2. Compliance Check Methods*

Each method within Compliance Manager is designed to check compliance of specific AWS services or policies:

- Check_compliance_feature: Helper method to update compliance status based on features found in each container dictionary.
- Check_iam_compliance: Verifies IAM password policies and updates compliance features accordingly.
- Check_s3_encryption_at_rest: Checks if S3 buckets have encryption-at-rest enabled and updates compliance features.
- S3_secure_data_acl: Checks S3 bucket policies related to access control lists (acls) and access permissions.
- Kms_compliance_audit: Verifies the use of AWS Key Management Service (KMS) for cryptographic key management.
- Check_cloud_trail_logging: Ensures that AWS cloudtrail is properly logging and monitoring AWS resource activities.

#### *3.2.3. Handling Errors and Responses*

Each method handles exceptions (specifically ClientError from AWS API interactions) to catch and record errors in the compliance checks.

The compliance status after checks is printed or updated in a structured format which can be further analyzed or reported.

#### *3.2.4. Usage*

The script initializes an instance of Compliance Manager and provides a structure (container) to hold the compliance check results.

The results from various checks can be executed and printed to assess compliance status across different AWS services.

### 3.3. Implementation, Testing and Validation

In the implementation phase armed with the design specifications we move forward to bring the code to life. Python scripts are utilized, making use of Boto3 and dotenv to establish connections, within the AWS realm. Security measures are put in place for credentials environments are set up. The Compliance Checker class is poised to fulfill its responsibilities by conducting compliance checks with each method call.

Approaching the end of implementation testing and validation become stages. Unit tests are meticulously created using pytest to evaluate each methods functionality across scenarios. Mock responses brought forth by the moto library move through the system ensuring its robustness without engaging AWS resources. These trials assess the frameworks strength and readiness to protect AWS realms from compliance issues.

## 4. Result

The logging results for compliance evaluation encompassed class labels, metrics, individual class analysis, and overall metrics (see table 1). Class labels included non-compliant, partially compliant, and compliant instances. Metrics such as precision, recall, and F1-score were employed to assess prediction accuracy, with support indicating the number of true instances in each class. Individual class analysis revealed varied performance: non-compliant instances achieved high precision and moderate recall, partially compliant instances exhibited low precision but perfect recall, while compliant instances showed poor performance with zero precision and recall. Overall metrics demonstrated a 60% accuracy rate, with macro and weighted averages indicating a balanced evaluation across all classes, with weighted averages reflecting the impact of class distribution on performance metrics.

**Table 1** Log Container Report Prediction **-** Performance Metrics for Compliance Classification Model

| | **Precision** | **Recall** | **F1** | **Support** |
|---|---|---|---|---|
| NON-COMPLIANT | 1.00 | 0.67 | 0.8 | 3 |
| PARTIAL COMPLIANT | 0.33 | 1.00 | 0.50 | 1 |
| COMPLIANT | 0.00 | 0.00 | 0.00 | 1 |
| ACCURACY | | | 0.60 | 5 |
| MACRO AVERAGE | 0.44 | 0.56 | 0.43 | 5 |
| WEIGHTED AVERAGE | 0.67 | 0.60 | 0.58 | 5 |

In the analysis of I AM Compliance, the evaluation of classlevel metrics reveals significant deficiencies in the model's performance across different compliance categories (see table 2). For non-compliant instances, precision, recall, and F1-Score all scored at zero, indicating a complete failure in correctly identifying this class. The support value of 3 highlights the total instances of this class in the test data. Conversely, Partially Compliant instances were entirely absent from the test set, resulting in zero values for precision, recall, and F1-Score. For instances labeled as Compliant, the precision of 0.25 suggests that only a quarter of the predictions were accurate, while a recall of 0.50 indicates that half of the actual instances were correctly identified. The F1-Score of 0.33 reflects a balance between precision and recall. Overall metrics paint a bleak picture, with an accuracy rate of 0.20, and both macro and weighted average metrics indicating inadequate performance.

**Table 2** IAM Report Prediction **-** Performance Metrics for Compliance Classification Model

| | **PRECISION** | **RECALL** | **F1** | **SUPPORT** |
|---|---|---|---|---|
| NON-COMPLIANT | 0.00 | 0.00 | 0.00 | 3 |
| PARTIAL COMPLIANT | 0.00 | 0.00 | 0.00 | 0 |
| COMPLIANT | 0.25 | 0.50 | 0.33 | 2 |
| ACCURACY | | | | 5 |
| MACRO AVERAGE | 0.08 | 0.17 | 0.11 | 5 |
| WEIGHTED AVERAGE | 0.1 | 0.2 | 0.13 | 5 |

The evaluation of the KMS Compliance Check revealed significant deficiencies across compliance categories (See table 3). Non-Compliant instances had both Precision and Recall scores of 0.00, resulting in an F1-Score of 0.00, indicating a complete failure in classification. Partially Compliant instances showed slightly better performance, with a Precision of 0.33 and perfect Recall (1.00), resulting in an F1-Score of 0.50. Conversely, Compliant instances were entirely misclassified, with both Precision and Recall at 0.00. Overall accuracy was 0.20, with macro and weighted averages emphasizing the impact of class distribution.

**Table 3** KMS Report Prediction **-** Performance Metrics for Compliance Classification Model

| | **Precision** | **Recall** | **F1** | **Support** |
|---|---|---|---|---|
| NON-COMPLIANT | 0.00 | 0.00 | 0.00 | 3 |
| PARTIAL COMPLIANT | 0.33 | 1.00 | 0.50 | 1 |
| COMPLIANT | 0.00 | 0.00 | 0.00 | 1 |
| ACCURACY | | | 0.20 | 5 |
| MACRO AVERAGE | 0.11 | 0.33 | 0.17 | 5 |
| WEIGHTED AVERAGE | 0.07 | 0.2 | 0.1 | 5 |

In the evaluation of S3 Compliance, examining the classlevel metrics provides insights into the model's performance across different compliance categories (see table 4). For noncompliant instances, the precision of 1.00 suggests that all predicted instances were accurate, yet this metric is skewed due to a low recall of 0.25, indicating that only 25% of actual non-Compliant instances were correctly identified. The F1Score of 0.40 reflects a balance between precision and recall. However, Partially Compliant and Compliant classes both exhibit zero precision, recall, and F1-Score due to either absence or misclassification of instances in the test set. Overall metrics indicate a low accuracy rate of 0.20, with macro-average metrics suggesting limited performance across all classes, and weighted-average metrics showing an improvement but still highlighting deficiencies.

**Table 4** S3 Report Prediction - Performance Metrics for Compliance Classification Model

| | **PRECISION** | **RECALL** | **F1** | **SUPPORT** |
|---|---|---|---|---|
| NON-COMPLIANT | 1.00 | 0.25 | 0.4 | 4 |
| PARTIAL COMPLIANT | 0.00 | 0.00 | 0.00 | 0 |
| COMPLIANT | 0.00 | 0.00 | 0.00 | 1 |
| ACCURACY | | | 0.20 | 5 |
| MACRO AVERAGE | 0.33 | 0.08 | 0.13 | 5 |
| WEIGHTED AVERAGE | 0.80 | 0.20 | 0.32 | 5 |

The evaluation of S3_ACL Compliance provides insights into the model's performance in classifying compliance instances (see table 5). For the non-compliant class, a precision of 1.00 indicates accurate classification of all predicted instances without false positives, while a recall of 0.50 suggests that only half of the true instances were correctly identified, leaving room for improvement. The absence of Partially Compliant instances in the test set results in zero precision, recall, and F1-Score for this class. Conversely, the Compliant class demonstrates perfect precision and a recall of 0.67, indicating accurate identification of two-thirds of the true instances. Overall metrics indicate a 60% accuracy rate, with a macro-average precision of 0.67 and a relatively low recall of 0.39 due to some classes being poorly detected. The weighted-average precision of 1.00 is skewed by perfect precision in NonCompliant and Compliant predictions, while the weightedaverage recall of 0.60 reflects the accuracy due to underrepresentation of some classes. The F1-Score of 0.75 combines precision and recall, weighted by support, providing a comprehensive measure of overall performance.

**Table 5** S3_ACL Report Prediction - Performance Metrics for Compliance Classification Model

| | **Precision** | **Recall** | **F1** | **Support** |
|---|---|---|---|---|
| NON-COMPLIANT | 1.00 | 0.50 | 0.67 | 2 |
| PARTIAL COMPLIANT | 0.00 | 0.00 | 0.00 | 0 |
| COMPLIANT | 1.00 | 0.67 | 0.80 | 3 |
| ACCURACY | | | 0.60 | 5 |
| MACRO AVERAGE | 0.67 | 0.39 | 0.49 | 5 |
| WEIGHTED AVERAGE | 1.00 | 0.60 | 0.75 | 5 |

## 5. Discussion

Upon reviewing the logging data, the model excelled in detecting instances of Non-Compliance with accuracy albeit, at a lower recall rate. The Partially Compliant category although showing recall due to its representation, suffered from poor precision. However accurately categorizing Compliant instances proved challenging for the model likely due to their occurrence in the dataset. To boost performance, it is advisable to rectify class imbalances by introducing instances of categories such as Compliant and fine tuning the model for improved predictive abilities across all classes.

In analyzing I AM Compliance, significant challenges emerge in classifying compliance instances, especially for Non-Compliant and Compliant classes. Non-Compliant cases pose a challenge with zero precision and recall indicating the models struggle in identifying them. On the hand while Compliant instances exhibit recall rates, low precision hints at an issue of over prediction resulting in false positives. To address these challenges effectively it is crucial to ensure a distribution of classes, fine tune model parameters for better class distinction and refine feature selection processes to encompass all factors. Implementation of these suggestions will enhance the model's precision and reliability in compliance classification, leading to decision making, in compliance management.

The assessment of S3 Compliance brings to light hurdles in categorizing compliance cases. The dominance of the Non category results in accurate but limited recall while the lack of Partially Compliant instances and minimal representation of the Compliant category worsen the imbalance. Moreover, achieving precision, for Noncompliant cases with low recall indicates potential false positives. To boost performance, it is advised to tackle class imbalances, test alternative models or parameters and fine tune feature engineering. These measures are essential for enhancing the precision and dependability of compliance classification.

An examination of S3_ACL Compliance uncovers insights into how the model performs across different compliance categories. Although there is precision in the Noncompliant category the low recall suggests that the model might be missing many true instances indicating room for improvement in accurately identifying non-Compliant cases. The absence of Partially Compliant instances in the test data complicates evaluating this class resulting in zero metrics being recorded. In contrast the Compliant category shows promising performance with precision and F1 score indicating identification of instances. To enhance model effectiveness several suggestions are put forward. Firstly, addressing the lack of Partially Compliant instances by increasing their presence in both training and testing datasets is crucial, for a evaluation.

Additionally enhancing the characteristics to guarantee the model can effectively distinguish between categories may boost classification precision. Finally adjusting the model with a focus, on improving recall for Non cases could also strengthen its capability to accurately spot instances within all compliance categories. Putting these suggestions into practice is crucial, for enhancing the efficiency and dependability of the compliance classification model.

### *Limitation*

Our compliance classification model has shown promise in identifying classes, but we need to acknowledge some limitations. One major challenge is the imbalance in class representation with an overabundance of Noncompliant instances and a lack of Partially Compliant instances in the test data. This imbalance affects performance metrics making it difficult for the model to generalize effectively across all compliance categories. Additionally relying heavily on features may restrict the model's ability to accurately differentiate between classes. Moreover, the models struggle with recall for Noncompliant instances, indicates potential missed detections highlighting the need for improvement in detecting true instances of noncompliance. Lastly due to classes like Partially Compliant instances being absent from the test data, a comprehensive evaluation of the model's performance across all compliance categories is hindered. To address these limitations and enhance the model's effectiveness and robustness in compliance classification strategies such, as increasing class representation, refining features and adjusting model parameters are crucial.

### *Recommendations*

- Recommendations for AWS Users

To enhance compliance with ISO 27001 and ensure robust security management, AWS users should adopt a comprehensive strategy that incorporates the use of automated API framework tools. Here are some key recommendations:

- Implement Continuous Monitoring and Auditing:

AWS users should deploy continuous monitoring and auditing tools to ensure real-time compliance with security standards. Tools such as AWS Config and AWS Security Hub can provide ongoing assessments of your environment, identifying potential non-compliance and enabling quick remediation.

- Enforce Strong Access Controls:

Ensure that IAM policies follow the principle of least privilege, granting only necessary permissions to users and roles. Regularly review and adjust permissions to prevent privilege creep and ensure that access is tightly controlled.

- Utilize Encryption for Data Protection:

Enable server-side encryption (SSE) for all data stored in S3 buckets using SSE-S3 or SSE-KMS. This ensures that data at rest is protected. Additionally, enforce the use of HTTPS for data in transit to safeguard against interception and man-in-the-middle attacks.

- Regularly Rotate Keys and Credentials:

Implement a policy for regular rotation of encryption keys and access credentials. This reduces the risk of compromised keys and ensures that only authorized personnel have access to critical data.

## 6. Conclusion

The evaluation of compliance classification models across various categories reveals significant insights into their performance. While these models excel at identifying Non cases, they face challenges with Compliant and Partially Compliant categories. The imbalance in class distribution poses an obstacle impacting the model's ability to generalize effectively. Moreover, depending on features and model, parameters can hinder its accuracy in distinguishing between classes. Furthermore, the models show weaknesses in recalling Non instances suggesting potential missed detections. To overcome these challenges and improve the model's efficacy it is essential to implement strategies, like increasing class representation, refining features, and adjusting model parameters.

## Compliance with ethical standards

*Disclosure of conflict of interest*

No conflict of interest to be disclosed.

## References

[1] APITier. (2024). How to Choose API Testing Tools & Frameworks.

[2] AWS. (2024). Cloud Security, Identity, and Compliance Products – Amazon Web Services.

[3] AWS. (2024). Encryption Cryptography Signing - AWS Key Management Service - AWS.

[4] AWS. (2024). Getting Started | AWS Key Management Service (KMS) | Amazon Web Services (AWS).

[5] AWS. (2024). New AWS Key Management Service (KMS) | AWS News Blog.

[6] AWS Security Blog. (2024). How to use KMS and IAM to enable independent security controls for encrypted data in S3.

[7] Blink. (2024). Checking S3 Bucket Encryption Compliance Across Your AWS Account.

[8] Chiregi, M., & Navimipour, N. J. (2018). Cloud computing and trust evaluation: A systematic literature review of the state-of-the-art mechanisms. Journal of Electrical Systems and Information Technology.

[9] Cloudflare. (2024). API Security Solutions.

[10] Cloudlytics. (2022). ISO 27001: 2022: 5 Steps for Cloud Systems to Comply Better.

[11] Databricks. (2024). Configure encryption for S3 with KMS.

[12] Datamation. (2024). 10 Best Cloud Compliance Tools of 2024: Expert Comparison.

[13] Hyperproof. (2024). Cloud Compliance Frameworks: What You Need to Know.

[14] ISO Council. (2022). Benefits of ISO 27001 Certification in the Age of Cloud Computing.

[15] Jennings, B., & Stadler, R. (2015). Resource management in clouds: Survey and research challenges. Journal of Cloud Computing.

[16] Kuyoro, S. O., Ibikunle, F., & Awodele, O. (2011). Cloud computing security issues and challenges. International Journal of Computer Networks, 3(5), 247-255.

[17] Manvi, S. S., & Shyam, G. K. (2014). Resource management for infrastructure as a service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications.

[18] Microsoft. (2024). Azure Policy Regulatory Compliance controls for Azure API Management.

[19] Microsoft. (2024). Web API Implementation - Best Practices for Cloud Applications.

[20] Mitchell, C. (2015). Privacy, compliance, and the cloud. Guide to Security Assurance for Cloud Computing.

[21] NIST. (2021). Evaluation of Cloud Computing Services Based on NIST 800-145.

[22] Palo Alto Networks. (2024). Cloud Visibility, Cloud Compliance & Cloud Governance.

[23] Postman. (2024). What Is API Governance? Best Practices & Getting Started.

[24] Postman. (2024). What Is API Governance? Best Practices & Getting Started.

[25] RapidAPI. (2024). What is API Governance? Governance Examples, Benefits, Best Practices.

[26] RapidAPI. (2024). What is API Governance? Governance Examples, Benefits, Best Practices.

[27] ScaleSec. (2024). Strategies for Enhancing Cloud Security and Compliance through Automation [Image]. ScaleSec. Retrieved from https://scalesec.com/blog/cloud-automation-tips-for-security-compliance/

[28] Singh, S., & Chana, I. (2017). Resource provisioning and scheduling in clouds: QoS perspective. Journal of Cloud Computing: Advances, Systems and Applications.

[29] Snyk. (2024). Cloud Compliance Tools Guide - Capabilities & Techniques.

[30] Solo.io. (2024). API Management Tools: Top Features & 8 Tools You Should Know.

[31] Sprinto. (2024). List of Compliance Framework (Complete Guide).

[32] Stoplight. (2024). API Security Management & Best Practices.

[33] Sysdig. (2022). Understanding ISO 27001 Compliance for Containers and Cloud.

## Appendices

| Control ID | Control Description | ApplyServer SideEncryption ByDefault | Password ReusePrevention | MFA | Hard Expriy | Max Password Age | Minimum Password Length | Expire Password | Secure Transfer Config | Key Management sevice | Access Restriction | User Management | Privileged Access | Secret Auth Management | Logging Monitoring | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A.13.1.1 | Network Security Management: S3 buckets should be configured to ensure secure data transfers to and from the service, adhering to the standard's requirement for managing network security. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | 100 |
| A.13.2.1 | Information Transfer Policies and Procedures: Policies for securing uploads/downloads and sharing of S3 data should | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | 100 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | be in place, aligning with controls on information transfer. | | | | | | | | | | | | | | | |
| A.10.1.1 | Cryptographic Controls: AWS KMS is central to managing cryptographic keys for data encryption, directly supporting the control requiring the use and management of cryptographic techniques and keys. | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | | 100 |
| A.10.1.2 | Management of Cryptographic Keys: Specifically addressing the lifecycle management of cryptographic keys, including generation, | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | | 100 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | distribution, storage, and destruction, which KMS facilitates. | | | | | | | | | | | | | | | |
| A.9.2.3 | Management of Privileged Access Rights: IAM is crucial for managing special access rights, ensuring that only authorized personnel have elevated access permissions, in compliance with this control. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | | 100 |
| A.9.4.1 | Use of Secret Authentication Information: IAM supports the management of secret authentication information (passwords, keys) through policies enforcing password complexity, rotation, and | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | 100 |

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | multi-factor | | | | | | | | | | | | | | | |
| A.12. 4.1 L | Logging and Monitoring: Integration of IAM with AWS CloudTrail ensures that logging and monitoring controls are met by recording and analyzing actions made on AWS resources. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 100 |