National College of Ireland

Open-Source Security and Confidentiality Framework for Multi-Cloud Environments

MSc Research Project
Cloud Computing

Anjali Sandeep Wagaskar
Student ID: x23340363

School of Computing
National College of Ireland

Supervisor: Dr. Punit Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anjali Sandeep Wagaskar
Student ID:	x23340363
Programme:	Cloud Computing
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr. Punit Gupta
Submission Due Date:	10/08/2025
Project Title:	Open-Source Security and Confidentiality Framework for Multi-Cloud Environments
Word Count:	3004
Page Count:	17

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:	Anjali Wagaskar
Date:	14th September 2025

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):	

Open-Source Security and Confidentiality Framework for Multi-Cloud Environments

Anjali Sandeep Wagaskar
x23340363

Abstract

The growing use of multi-cloud has created major concerns in terms of security due to its instability and inconsistency. This thesis shows a novel, strategic integration and implementation of an open-source security ecosystem consisting of four core domains which are directly mapped to the capabilities of the tools, the Open Policy Agent (OPA) for centralised access control, secret management with HashiCorp Vault, scalable object storage with MinIO, and real-time threat detection through Falco. The framework implements role based fine grain, access policies and dynamically issues credentials and monitors storage operations to detect and alert unauthorised activities. Experimental verification confirms the functional rightness of the framework, proving policy enforcement, secure management of credentials and on-going intrusion detection. Low latency and high throughput are shown to be the performance benchmarks of policy evaluation and secret retrieval, which establishes the possibility of the framework being practical. The scalability, policy flexibility and alert integration limitation are discovered. In terms of improving the state of open-source cloud security tools, this study delivers a vendor-neutral security solution that protects multi-cloud systems, addressing the short term needs of cloud security on an industry level.

Keywords: Multi-cloud security, Access control, Open Policy Agent (OPA), HashiCorp Vault, MinIO, Runtime threat detection, Falco, Secrets management, Role-based access control (RBAC), Object storage security, Open-source security framework, Policy enforcement, Performance evaluation

1 Introduction

The rapid growth in digitisation of industries has led to an increasing adoption of multi-cloud strategy. Almost 86% of organisations have adopted a multi-cloud approach in 2025 (Flexera; 2025). Organisations also use the advantages of various cloud service providers to achieve rapid marketing, cost reduction and agility to cope with the changing technologies. This approach provides seamless benefits like resilience and vendor independence, but it expands the security threat surface. Much to its credit, it led to the emergence of biggest challenge involving the need to support strong security in the various cloud platforms. According to 2024 state of multi-cloud security risk report, organisations experienced at least a data exposure incident Microsoft (2024). Recently, cyber threats has

escalated, targeting cloud workloads at scale with AI-powered attacks and automated exploitation toolsCrowdStrike (2024)IBM Security (2025). During the same time regulatory requirements has placed additional pressure on organisations.

In environments like this security is in fragmented, as each provider has different tools, policies and monitoring mechanisms. This leads to a complex security management, lack of protection and capabilities on interoperability. Existing solutions are typically focused on a single aspect of issue and many of these tools are proprietary and expensive, restricting flexibility scalability, and adaptability.

In current research and industry practices, there is absence of an integrated, open-source, vendor-neutral framework that unifies security risks like threat detection, automated policy enforcement, secrets management, and secure multi-cloud storage within a single system that will fit together. lack of automation in security enforcement and indecent response, there are many solutions but still requires manual handling. These gaps underscore the need for unified, open-source approach that will fulfill interoperability, automation and Regulation-aligned security.

1.1 Motivation

Due to increased adoption of multi-cloud environments by organisations interested in flexibility, scalability and cost-consciousness of operations, data and services security within various cloud platforms has become difficult. It is common that traditional security solutions do not support a consistent move dynamic low-latency security and access control in this distributed environment. Also, increasing complexity of cyber risks requires real-time detection and response capabilities that is supported in a multi-cloud environment. The rationale behind the project is the necessity to create a unified, efficient, scalable system of security that can combat these challenges and enable secure management of secrets as well as enforce flexible policies and determine anomalies as early as possible to secure critical assets of the cloud infrastructure.

1.2 Research Question

What are the primary security challenges in multi-cloud environments, and how can open-source framwork be integrated to address them over proprietary solutions?

1.3 Research Objectives

This work, focuses on a novel open-source security framework, that integrates four open-source security tools specifically for multi-cloud environment. Tool the framework strategically integrates:

- Open Policy Agent (OPA) : automated policy enforcement.
- HashiCorp Vault: secure secrets management and credential protection.
- MinIO: multi-cloud object storage.
- Falco: Real time threat detection.

- all of this tools are widely recognised for their platform independence and ability to be deployed across different cloud providers and on-premises environments. This solution

overcomes the need of a unified, vendor-neutral, and fully automated security solution, it is capable of operating across different cloud platforms. This design enabling organisations to enhance threat mitigation, enforce consistent policies, and safeguard sensitive assets by emphasising interoperability, scalability, and cost-effectiveness.

The framework design inherently supports multi-cloud deployment, the implementation and evaluation have been conducted within a controlled AWS environment. This serves as a proof of concept, demonstrating the effective integration and operation of these components in a cloud setting. The architecture is structured into distinct layers for event detection, policy enforcement, secrets management, and secure storage, all interconnected via open standards to ensure interoperability, scalability, and cost-effectiveness. A functional prototype demonstrates end-to-end integration with dynamic runtime behavior, token-based access control, and secure data handling in a simulated industry scenario.

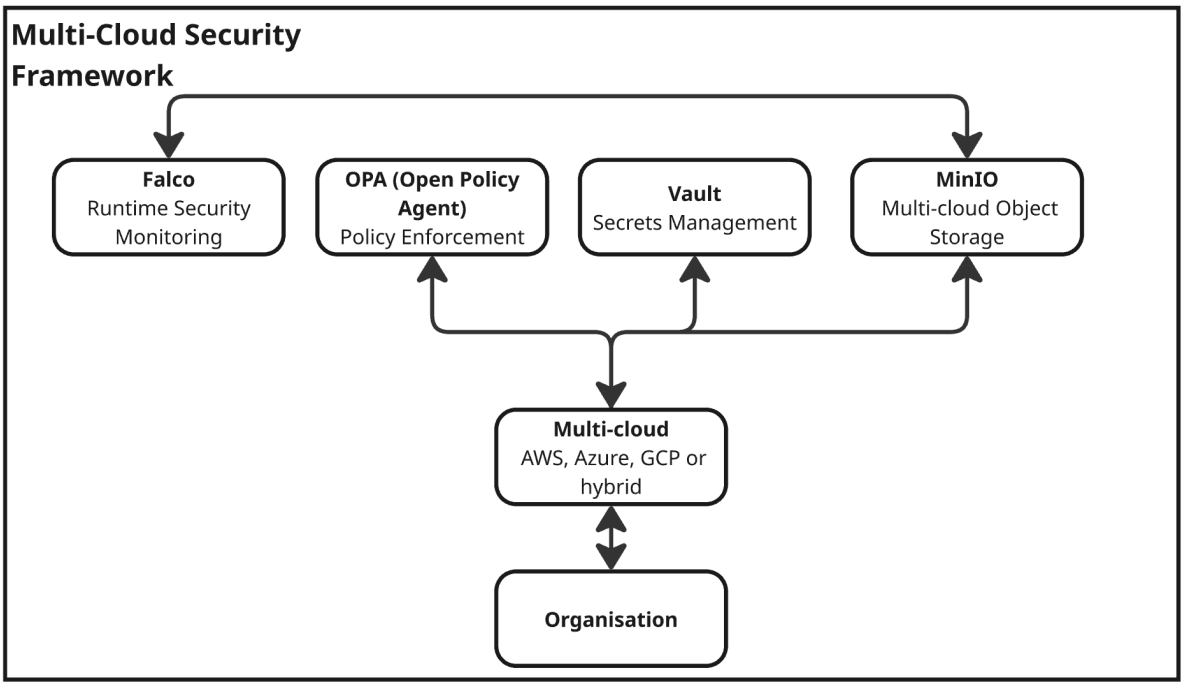

Figure 1: This image shows the frameworks overview

1.4 Contributions

- Proposes and implements a novel open-source security and confidentiality framework specifically for multi-cloud environments.
- Strategically integrates of security tools.
- Emphasises scalability and cost-effectiveness to meet organisational needs.
- Enables enhanced threat mitigation, consistent policy enforcement, and protection of sensitive assets without vendor lock-in.

2 Related Work

2.1 Multi-Cloud Security Challenges and solutions

Security and privacy in multi-cloud and hybrid cloud environments: Challenges, strategies, and future directions: This paper identifies key challenges like data breaches, misconfigurations, and compliance issues also discusses various strategies to address these concerns. Proposes unified security frameworks but mostly conceptual or proprietary. Recommends enhanced monitoring but no specific real-time detection mechanisms. Mentions unified policies, but no concrete policy enforcement mechanism that will show how to technically enforce those rules in practice(Ali et al.; 2025).

Challenges and Best Practices in Cloud Security Across Multi-Cloud Environments: This research explores some key security challenges inherent in multi-cloud deployments. Those are data fragmentation, inconsistent security policies, identity and access management complexities, and compliance across diverse platforms. It lack in practical open-source multi-tool integration and its conceptual and lacks demonstration(Cate; 2025).

Security Challenges in Multi-Cloud Environments: Solutions and Best Practices: This research analyses the security challenges, emphasises how these challenges increase organisational vulnerability and complicate unified security management. It focuses on high-level best practices and commercial tools, does not provide an integrated open-source framework(Madanan et al.; 2024).

Security Challenges and Solutions in Multi-Cloud Environments: This paper examines the security challenges in multi-cloud environments and it proposes mitigation strategies. Emphasises high-level best practices and commercial tools, recommends SIEM and behavior analytics for detection, mentions IAM but does not address dynamic secrets management, suggests unified policy but lacks dynamic policy enforcement tools(Paul; 2024).

Systemic Risk and Vulnerability Analysis of Multi-Cloud Environments: This study performs a risk and vulnerability analysis to identify attack vectors from software, hardware, and the network, as well as interoperability security issues in multi-cloud environments. The authors employ the STRIDE and DREAD threat modeling methodologies to identify and assess potential threats across six key attack vectors. Absence of specific tool-based threat detection, limited focus on open-source, vendor-neutral tools and no secure object storage solutions(Reece et al.; 2023).

2.2 Frameworks and Integrated Architectures

Cyclone Description and Methodology: Built an open-source federation-based cloud management tools that facilitates multi-cloud orchestration, security through IAM and SSL encryption. Illustrated through a prototype in several clouds. Mainly centered on deployment and orchestration with the security being an afterthought not threat detection, secrets automation, or unified storage security integration(Slawik et al.; 2016).

A Multi-Cloud DevOps Framework to be Agile and Efficient Description and Methodology: Offers a DevOps pipeline that would undertake cross cloud deployments, evaluation of real-time operational environments costs/performance and monitoring. Gap: The issue of security is handled at a conceptual level-monitoring is presumed, but no real-time detection or automatic policy enforcement tooling is either applied or tested(Pochu et al.; 2024).

Zero Trust Multi-Cloud Networking based on Open-Source Micro-Segmentation Description and Methodology: Design and deployment of zero trust networking with micro-

segmentation tools made available as open-source software, tested with simulated multi-cloud network traffic. Gap: Only network layer is covered, policy enforcement (not limited to network ACLs), secrets management, storage protection, and detection with Falco is not included(Arora and Hastings; 2024).

Secure Multi-Cloud Integration Framework Integration with the Use of Honeypots: Distributing Defensive Designs and implements a multi-cloud honeypot to track and capture behaviors of an attacker in a real-time cross-cloud strategy. Centralised logging and alerting are a part of the framework. It is innovative in itself in the field of threat observation, but it is reactive in its way of collecting such data through honeypots as opposed to integrated enforcement. It lacks security checks of active policies, secret management, and secure layers of storage(Alyas et al.; 2022).

2.3 Industry Standards and Frameworks

A number of implemented standards and guidelines offer security best practices that can be used in multi-cloud environments, although, the majority of them are of a high level and do not stipulate specific, automated actions of implementation.

NIST SP 800-61 Revision 2 (Computer Security Incident Handling Guide) and the expected Revision 3 include guidance on incident response in detail with a primary focus on incident response procedures based on identification, containment, eradication, and recovery of security incidents. Although they are mainly oriented at incident handling, their principles can be applied to multi-cloud policies to encourage similar incident management processesCichonski et al. (2012)Nelson et al. (2025).

This guide, a companion to the Center for Internet Security CIS Controls V8.1, expands the universe of core security controls to cloud environments, providing cloud-specific workload securing measures. On the same note, CIS Benchmarks offer configuration baselines on cloud providers like AWS, Azure and Google cloud that are grounded in a consensus configuration to mitigate the risks of misconfigurationfor Internet Security (2023).

The Cloud Security Alliance (CSA) offers best practices when working in the cloud, which not only explain the advantages of multi-cloud adoption such as flexibility, but also the configuration and visibility risks (Amroussi, 2021). NIST Multi-Cloud Security Public Working Group (MCSPWG) and CSA emphasise the importance of multi-vendor uniform visibility and policy control.Cloud Security Alliance (2015).

Despite the fact that existing frameworks can be used to theoretically implement multi-cloud security, they tend to be platform-restrictive and cannot be enforced quickly in real-time, or across any technology platform. This shows the importance of the open-source solutions that fill in the gap between best practices and practical and uniform security with different cloud providers and that is what the proposed Open-Source Security and Confidentiality Framework is supposed to accomplish.

2.4 Open-Source Tools for Multi-Cloud Security

HashiCorp Vault: is a secret management platform that allows the secure storage and access the storage of sensitive data on different cloud environments. It has the feature of dynamic secrets, identity-based access, and audit logging that play a vital role in ensuring the security of multi-cloud environments is maintained.

Open Policy Agent (OPA): is a policy engine which facilitates policy-based security in a centralised manner with consideration of context across multiple services and platforms. On multi-cloud environments, OPA promotes the uniformity of the applied policies which increases security posture.

Falco: ironically it is one of the tools to provide runtime security in that it watches system calls and notices anomalies in real-time. It establishes an insight into possible threats, and that is pivotal to security in various cloud infrastructures.

MinIO: is an open source object storage server that supports Amazon S3 APIs. It enables the organisations to utilise multi-cloud environments to handle and store unstructured content and is scalable and flexible.

2.5 Summary of Gaps

Existing works have explored multi-cloud threats but they are often limited in scope: being either abstract (theoretical frameworks lacking practical application), proprietary (commercial services with limited interoperability), or too fixated (focusing on a single threat domain such as IAM, storage, or detection). The existing research and industry standards define the following multi-cloud security issues: data fragmentation, difference in policy, identity management, and compliance. Frameworks and commercial tools offer best practices, but are usually high-level, platform-specific or do not offer real time detection and adaptive policies. Open-source projects, such as Vault, OPA, Falco, and MinIO, and multipurpose each solve discrete components of the multi-cloud security stack: secrets management, policy enforcement, run-time threat detection and storage but there is no integrated open-source framework available that solves the multi-cloud security stack. It is this gap that drives a unified, technology-agnostic, open-source security framework that is both compatible with the best practices and practical security implementation in multi-cloud environments.

3 Methodology

This study adopts a methodological pattern to conduct an analysis that will assess the strengths and weaknesses of an integrated security model to multi-cloud environments.

3.1 Research Approach

The approach that was used in the research was the design science research methodology, the goal of which is the creation and evaluation of an artifact, in this instance, the security enforcement structure. The performance is assessed with an accent on the validation of security workflow and performance measured in ideal laboratory conditions.

3.2 Security Workflow

Reconciliation The method that is studied validates the end-to-end security exercise provision process by exercising simulated user access demands and ensuring that: Access decisions are according to set policies, Secrets are also retrieved through granted authorisation in a secure manner. Object storage operation is carried out with valid credentials. Another use of runtime monitoring is that it identifies suspicious activities as they occur.

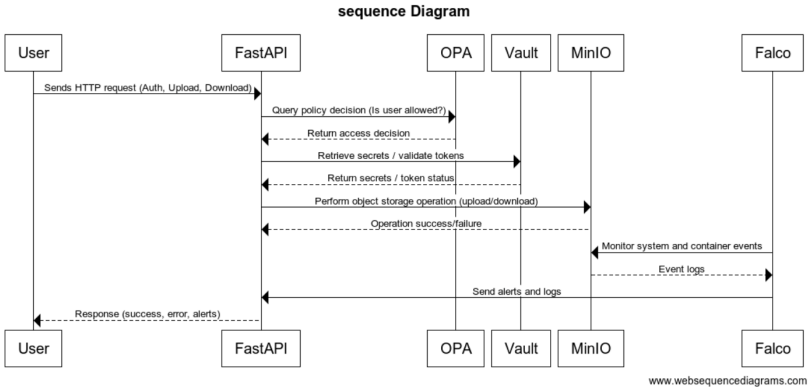

Figure 2: Sequence Daigram: Displaying workflow

3.3 Performance Evaluation

Performance benchmark was done where the focus was on the latency, throughput and the success percentage of the crux of the modules dealing with policy evaluation, secret retrieval and storage. These measures can inform the response of the framework and feasibility of its working.

3.4 Security Testing

The system was tested on its effectiveness of detecting and alerting of unauthorised access or suspicious activity in a simulated scenario that involved testing on the effectiveness of detecting intrusions at runtime.

3.5 Data Analysis

The strengths and limits of the framework were evaluated in terms of performance tests and security validation in the form of quantitative data analysis of performance tests and qualitative findings of the validation. Compare and contrast was done with the findings of other works in the literature that have been done to facilitate the contributions made.

4 Design Specification

The system being proposed is a multi-cloud security system that reflects on the challenges of policy compliance, secrets management, aware response to threats, and storage of secure data in its distributed cloud environment. The architecture is based on the modular and layered design to guarantee scalability, interoperability and vendor neutrality with good security assurance.

4.1 Overview of architecture

The framework is included in four main layers and one parallel in monitoring:

User Interaction Layer -This part is implemented with FastAPI that serves as the mechanism between administrators and users as a pivotal point. It does authentication, policy request, file operations and exposes APIs with REST to integrate with the other system.

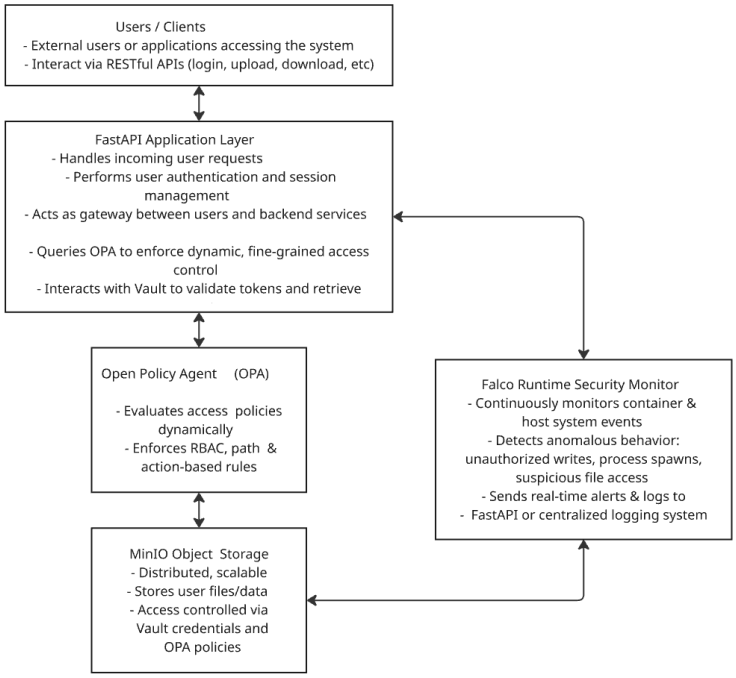

Figure 3: Architecture Diagram

Policy Enforcement Layer - Open Policy Agent (OPA)is used to enforce Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) policies using this layer. It compares and contrasts incoming requests with an existing Rego policy list and provides authorisation decisions.

Secrets Management Layer - is managed by HashiCorp Vault and securely stores, and retrieves sensitive credentials (e.g., MinIO access keys). The storage of secrets is done through encryption and can only be read through validated API calls.

Secure Storage Layer - Deployed using MinIO, S3 compatible object storage, a top encrypted storage and per user bucket isolation, to support multi cloud deployment.

Threat Detection Component - It is implemented using Falco, which is to monitor containerised workloads in real time and alert on suspicious behavior (e.g., unexpected system calls, file changes).

The elements interact through RESTful APIs over TLS-secured links, which guarantees sensitive and integrity of information passed on.

4.2 Workflow

The framework works in the following way:

The FastAPI gateway receives a request created by a system or a user (e.g., upload file, download file, view data).

FastAPI also verifies the request after which it submits it to OPA to evaluate the policy.

When authorised FastAPI requests the credentials required to be accessed in Vault and retrieves as secured credentials and connects to MinIO to store the information.

Falco also runs concurrently and keeps watch of the runtime environment and writes or notifies on any detected anomalies.

4.3 Major Demands

The requirements that the system was to meet included the following:

- Open-Source: The framework should be open-source
- Vendor Neutrality: This should be able to work in heterogeneous clouds.
- Secure Secrets Handling: No secrets will be stored in plaintext format and all access is controlled and auditable.
- Real-time Threat Detection: The system should alert and indicate of suspicious activities in real-time.
- Fine-Grained Policy Control: Enables both RBAC and ABAC models of flexible access control.
- Scalability and Performance: Should support concurrent requests and has minimal latency.

4.4 Originality and Addition

Though the tools included in this framework are independent of each other, their combination that forms a single, multi-layered architecture of security in multi-cloud ecosystems is a new development. The framework is partly different to all the other cloud-specific solutions as it provides high level of security and cross-compatibility across all clouds.

5 Implementation

The implementation step entailed the deployment and integration of the main open-source modules of the suggested multi-cloud security system on actual Amazon web services (AWS) EC2 virtual machines, which emulated real multi-cloud infrastructure and did not use containers. A total of four different tools, Open Policy Agent (OPA), HashiCorp Vault, MinIO, and Falco, were installed and configured separately on individual EC2 instances based on Ubuntu 22.04 LTS and Linux to mimic as much as possible a production environment in a multi-cloud setting.

FastAPI was written in python to serve as a central API gateway, and communication hub, coordinating services. Also, OPA was set to use custom Rego policies to dynamically control access on a role and attribute basis. Sample Rego policy that is used for demo purpose:

Rego Policy:

```
package main

default allow := false

roles := {
 "manager": {"allowed_endpoints": [
                {"path": "/v1/secret/myapp", "actions": ["read", "create", "update",
```

```
                {"path": "api/login/token", "actions": ["read"]},
        ]},
 "employee": {"allowed_endpoints": [
                {"path": "/v1/secret/myapp", "actions": ["read", "create"]},
                {"path": "api/login/token", "actions": ["read"]},
        ]},
 "admin": {"allowed_endpoints": []},
}

allow if {
 is_admin
}

has_path_perms if {
 ae := roles[input.roles[_]].allowed_endpoints[_]
 not endswith(ae.path, "*")
 input.path == ae.path
 input.action in ae.actions
}

has_path_perms if {
 ae := roles[input.roles[_]].allowed_endpoints[_]
 endswith(ae.path, "*")
 startswith(input.path, trim_suffix(ae.path, "*"))
 input.action in ae.actions
}
```

The installation of HashiCorp Vault was performed on a dedicated VM and initiated safely to store the MinIO credentials and to keep the API tokens. Privacy was limited by vault policies, and FastAPI services received authentication tokens in order to access secrets. An example of Vault policy:

```
path "secret/data/minio/*" {
  capabilities = ["read"]
}
```

MinIO was also run in distributed mode on many VMs, attached to a dedicated storage device, delivering object storage to make it scalable, isolate buckets and enable server-side encryption. Configuration was performed through environment variables and access policies by simply using the MinIO command line tool (mc). The MinIO client (mc) was configured to communicate with the MinIO server. Root user credentials were set via environment variables on each instance for secure access.

On the same VMs, Falco was installed as a host based runtime security agent, able to monitor system calls, in real-time generating security alerts according to custom parameters/ rules set to identify abnormal behaviors. A RESTful APIs interface over HTTPS and TLS certificates configured manually on individual servers was used to support all the inter-service communication based on the need to guarantee the security of the data in transit and the confidentiality and integrity of the information.

Falco Rules:

```
macro: minio_dir
  condition: fd.name startswith /mnt/data/minio
```

-
```
  macro: proc_name_exists
  condition: proc.name != "" and proc.name != "unknown"
```

-
```
  macro: spawned_process
  condition: evt.type in (execve, execveat)
```

-
```
  rule: Unauthorised write to MinIO directory
  desc: Detect any unauthorised write operation to MinIO bucket files
  condition: >
    minio_dir and open_write and proc_name_exists
    and not proc.name in (minio)
  output: >
    [FALCO] Unauthorised write to MinIO directory detected!
    user=%user.name command=%proc.cmdline parent=%proc.pname
    pcmdline=%proc.pcmdline file=%fd.name program=%proc.name
    gparent=%proc.aname[2] ggparent=%proc.aname[3]
    container_id=%container.id image=%container.image.repository
  priority: ERROR
  tags: [minio, filesystem, write, security]
```

-
```
  list: forbidden_shell_binaries
  items: [ find ]
```

-
```
  list: allowed_shell_binaries
  items: []
```

-
```
  rule: Shell binary spawned process
  desc: >
    Detect when forbidden shell binary (e.g., find) spawns a child process.
  condition: >
    proc.pname in (forbidden_shell_binaries) and spawned_process
    and not proc.name in (allowed_shell_binaries)
  output: >
    Shell binary spawned process other than itself
    (user=%user.name parent=%proc.pname pcmdline=%proc.pcmdline
    gparent=%proc.aname[2] container_id=%container.id
    image=%container.image.repository)
  source: syscall
  priority: NOTICE
  tags: [process, mitre_execution]
```

6 Evaluation

The assessment was aimed at the evaluation of the effectiveness, performance, and spheres of security solidity of the proposed multi-cloud security framework. The main goals were to calculate (1) whether the system achieved its expected use of enforcing fine-grained access control, securely managing secrets, detecting threats in real time and compatibility across multi-cloud environments. This evaluation metrics (latency, throughput, detection accuracy, resource utilisation) shows the direct connection with both operational and functional goals of the multi-cloud environment. It evaluates the matrics that are important in real world needs: Organisations need high performance (low latency, high throughput) with robust security (good detection, effective enforcement) whilst remaining highly efficent(low resources used).

In particular, four combined parts were examined:

- HashiCorp Vault is a secrets management tool.
- Open Policy Agent (OPA) - policy enforcement.
- MinIO secure object storage.
- Falco intrusion detection was explored when it comes to detecting violation of policy in real-time.

Benchmarks In a controlled low-latency setting, Python scripts were used to send requests or file operations to a given service. Each of the tests was performed one after the other to get raw responsiveness devoid of network congestion influence.

```
(venv) voclabs:~/environment $ python benchmark-vault.py
Completed 100 requests in 0.3196 seconds.
Successful requests: 100
Failed requests: 0
Average time per request: 3.20 ms
(venv) voclabs:~/environment $ python benchmark-opa.py
Completed 100 requests in 0.2984 seconds.
Successful: 100
Failures: 0
Average time per request: 2.98 ms
(venv) voclabs:~/environment $ python benchmark-minio.py
Uploaded 100 files in 1.62 seconds.
Average upload time per file: 0.0162 seconds.
Downloaded 100 files in 0.49 seconds.
Average download time per file: 0.0049 seconds.
Listed 100 objects in bucket 'test-bucket'.
Listing files took 0.0390 seconds.
(venv) voclabs:~/environment $
```

Figure 4: Test Scenarios of the tools integrated

Table 1: Performance Benchmark Summary

Comp.	Op.	Ops	Time (s)	Lat. (ms)	Thr. (ops/s)
Vault	Read secret	100	0.3196	3.20	∼313
OPA	Policy eval.	100	0.2984	2.98	∼335
MinIO	Upload	100	1.62	16.20	∼61
MinIO	Download	100	0.49	4.90	∼204
MinIO	List objects	100	0.0390	0.39	∼2564

The security effectiveness is falco detection: A controlled test was done where unauthorised process (sudo tee) tried to write to /mnt/data/minio repeatedly7. Falco produced alerts when each of many unauthorised attempts, even with high consecutive writing (100+ operations in milliseconds)6. There were no infiltration attempts without detection and this verifies the detection of all activities in real-time and zero false negative incidents in this case.

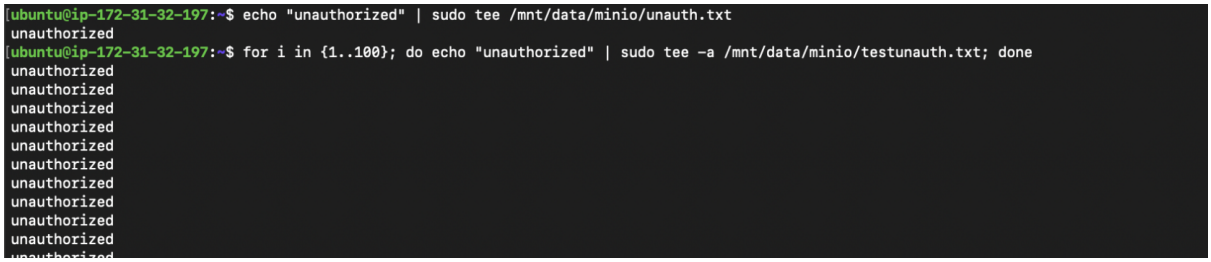

Figure 5: Test Scenarios of falco unauthorised access.

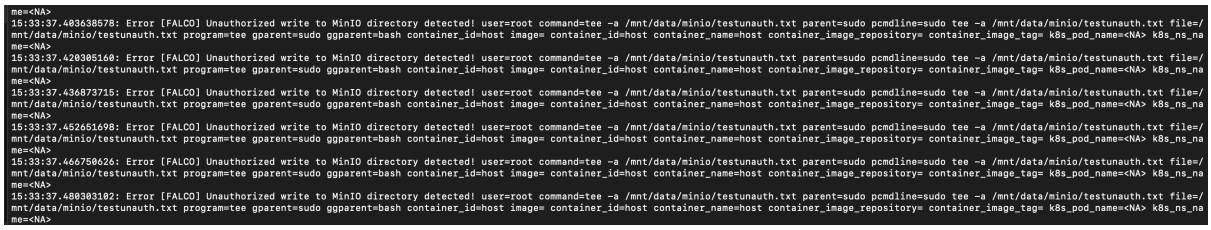

Figure 6: Test Scenarios result falco sending real-time alerts.

Analysis Vault supported 100 secure secret lookups within less than 0.32 seconds with an average latency of 3.20 ms making it appropriate to work in real time access control. OPA did policy evaluations even more quickly 2.98 ms per request (7% quicker than Vault), which is an indicator of its lightweight, in-memory mode of policy evaluation. MinIO has high download throughput (204 ops/s) and in particular, fast listing of objects (2,500+ ops/s), but uploads are much slower (61 ops/s) since this involves the write overhead of storage. Falco was able to achieve real-time detection so that it detected all unauthorised attempts to write without missing or delay.

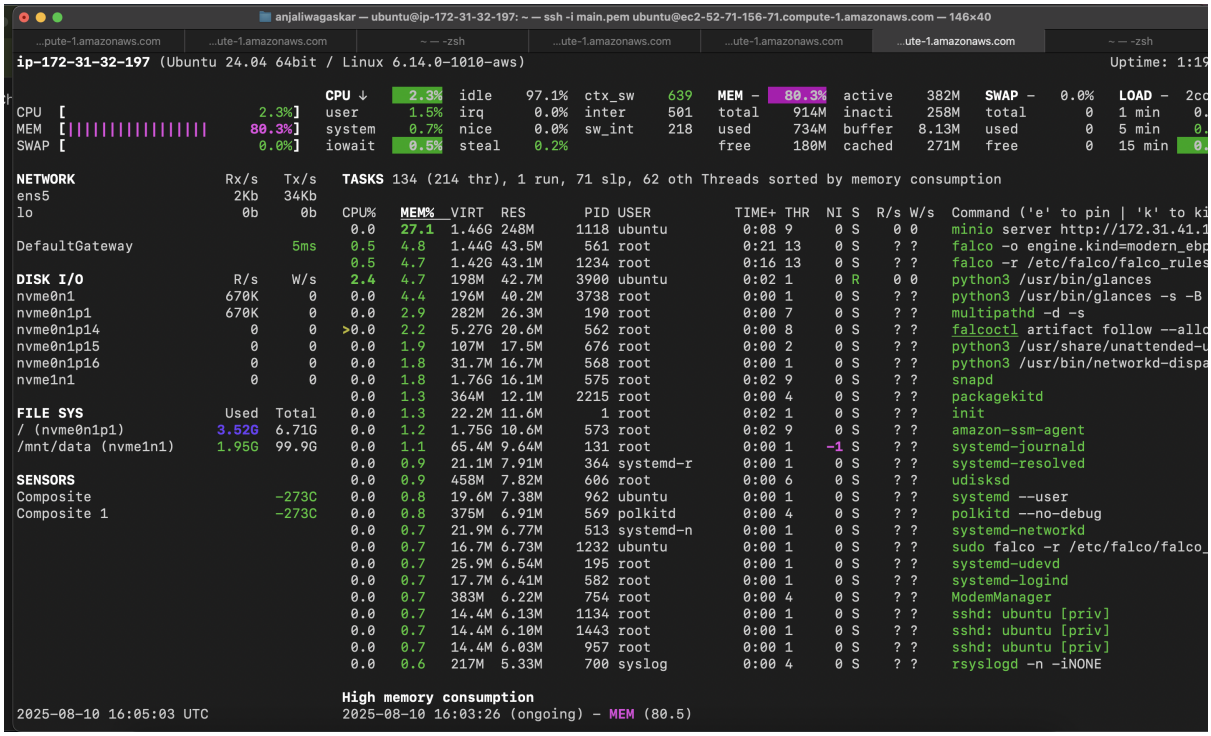

Figure 7: The screenshot shows the server's real-time performance metrics using the glances monitoring tool while running the integrated framework

6.1 System Resource Utilisation

CPU Usage: It is justifiable that the CPU load is low at 2.3% that shows that the security stack is not a heavy computational load when used normally.

Memory Utilisation: The general memory utilisation is 80.3% which is mostly because of active security services. The most consuming are:

minio server (27.1% MEM) -storage service

falco operations (45 MEM total) 4%-5% MEM per run time threat detection

python3 / glances or the overhead of monitoring tool

Disk I/O: There is very little active reads/writes on the disks except nvme0n1p1 which experiences stable read/writes on the nvme server storage.

Network Latency: Gateway default latency is 5ms and is sufficient in near real-time policy management and alerting.

Processes: They have all important security services (MinIO, Falco) that are running and maintaining within operating limits.

This shows that open-source multi-cloud security framework could be run with low CPU overhead, however, sufficient provisioning of memory is required to host multiple security agents that will be running continuously.

6.2 Security Workflow Validation

Along with performance testing, the end-to-end security enforcement workflow used by the integrated framework was also tested. The created flow works in the following way:

- Access Request: An end-user tries to access an operation on MinIO bucket.

- Policy Evaluation (OPA): OPA verifies the request with centralised role-based access policies.
  - In case the user is authorised to the requested path, OPA confirms a positive decision to Vault.
  - In the case of an unauthorised user, OPA rejects access and returns immediately.
- Secrets Management (Vault): After a positive evaluation by the OPA, Vault supplies the requisite credentials to the API in order to access MinIO.
- Object Storage Operation (MinIO): The API makes the invoked bucket or file operation with Vault credentials.
- Runtime Threat Detection (Falco): Falco is an active threat scanning package on the MinIO directory.

Some processes that do not have the permission to read, write, or modify files directly, Falco returns an alert (the example is located in Figure 6) and prevents any further operations.

6.3 Discussion

The experiments it conducted proved that the integrated framework, which comprises OPA to control the access and secrets management system Vault, MinIO to store the data, and Falco to identify the threats during runtime, operates as expected. The performance benchmarks demonstrated low latency of OPA and Vault procedures, and MinIO allowed acceptable speeds to fulfil the duties of an object store.

There were however certain limitations as seen in the evaluation:

- Time bounds: Only small-volume, sequential requests are used during the experimentation, therefore, it is unknown how the system will scale and perform when peak concurrency occurs. Other studies note that under heavy load Vault and other tools can become a bottleneck.
- Static Policy Model: Presently, lack of flexibility through use of static RBAC policies is an issue. As proposed in earlier works, more dynamic, attribute based policies are required in the modern cloud settings.
- Secret Management: Vault was deployed without dynamic secret rotation or just in time credentials, which can help mitigate the security impact and compliance.
- Detection Integration: Falco alerts were not built into automated incident response or centralised monitoring devices, which reduces the efficiency of the framework in timely serving the threats.
- Testing Environment: Low latency testing could be conducted under controlled lab conditions that might not resemble multi-cloud real-life set-ups that are characterised by more variability.

The design is good overall for proof of concept and small scale deployment and requires some changes to support large scale dynamic usage. It is suggested that future work should comprise testing of scalability, use of dynamic policies, more sophisticated secret lifecycle management, integration with SIEM and automated response tools, and experimentation with realistic cloud environments.

7 Conclusion and Future Work

This thesis gave a holistic security architecture involving Open Policy Agent (OPA), HashiCorp Vault, MinIO, and Falco to implement access control, control secrets, securely store objects and detect run-time threats within a multi-cloud environment. The framework has been proven to be correct in the execution of its operations, as the experimental assessment confirmed the framework operates in the manner detailed and exhibits great accuracy in policy enforcing, credential provisioning and real-time intrusion detection with low latency.

The toughest part of this research was the interface of the heterogeneous tools in a coherent, interoperable environment, together with providing the desired security and performance guarantees. This interoperability was achieved through designing a central API gateway to co-ordinate the actions. The technical difficulty was to achieve security enforcement rules and performance that is suitable for real-world multi-cloud use.

Nonetheless, the experiments also had shortcomings involving the capacity of scaling, flexibility of policy, and administration of its secret lifecycle, and its integration with alerts. These will be important when deploying the framework in large scale dynamic cloud environments.

Future work will be dedicated to further increasing scalability testing with high concurrency load, the usage of dynamic attribute-based policies, the introduction of automated secret rotation, and the use of just-in-time credentials, and also the coalescing of Falco alerts with centralised security monitoring and automated incident response. Also, the real-life mult-cloud implementation situations will be tested to check the performance and security in a distributed multifaceted environment.

This further evolution will make the framework more robust and applicable and that will lead to the establishment of stronger open-source security systems to multi-cloud environment.

References

Ali, S. et al. (2025). Security and privacy in multi-cloud and hybrid cloud environments: Challenges, strategies, and future directions, *Computers & Security* **157**: 104599.

Alyas, T., Alissa, K., Alqahtani, M., Faiz, T., Alsaif, S., Tabassum, N. and Naqvi, H. (2022). Multi-cloud integration security framework using honeypots, *Mobile Information Systems* pp. 1–13.

Arora, S. and Hastings, J. (2024). Microsegmented cloud network architecture using open-source tools for a zero trust foundation, *2024 17th International Conference on Security of Information and Networks (SIN)*, IEEE, p. 1–8.

Cate, M. (2025). Challenges and best practices in cloud security across multi-cloud environments, *ResearchGate* .

Cichonski, P., Millar, T., Grance, T. and Scarfone, K. (2012). Computer security incident handling guide.

Cloud Security Alliance (2015). Security guidance for critical areas of focus in cloud computing, version 3.0. Practical roadmap for securing cloud operations; foundational cloud security guidance.

CrowdStrike (2024). 2024 Global Threat Report: AI-Driven Attacks and Cloud Exploits, *Technical report*, CrowdStrike.

Flexera (2025). State of the Cloud Report 2025, *Technical report*, Flexera.

for Internet Security, C. (2023). Cis controls v8.1 cloud companion guide. CIS Cloud Security Companion Guide.

IBM Security (2025). 2025 Cost of a Data Breach Report: AI and Cloud Security Risks, *Technical report*, IBM Security.

Madanan, M., Patel, P., Agrawal, P., Mudholkar, P., Mudholkar, M. and Jaganraja, V. (2024). Security challenges in multi-cloud environments: Solutions and best practices.

Microsoft (2024). 2024 State of Multicloud Security Risk Report, *Technical report*, Microsoft.

Nelson, A., Rekhi, S., Souppaya, M. and Scarfone, K. (2025). Incident response recommendations and considerations for cybersecurity risk management: A csf 2.0 community profile.

Paul, A. L. (2024). Security challenges and solutions in multi-cloud environments, *International Journal of Cloud Computing and Services Science* **13**(2): 1–12.

Pochu, S., Nersu, S. and Kathram, S. (2024). Multi-cloud devops strategies: A framework for agility and cost optimization, *Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023* **7**: 104–119.

Reece, M., Lander, J., Stoffolano, M., Sampson, A., Dykstra, J., Mittal, S. and Rastogi, N. (2023). Systemic risk and vulnerability analysis of multi-cloud environments.

Slawik, M., Zilci, B., Demchenko, Y., Baranda, J., Branchat, R., Loomis, C., Lodygensky, O. and Blanchet, C. (2016). Cyclone unified deployment and management of federated, multi-cloud applications.