American Journal of Smart Technology and Solutions (AJSTS)

ISSN: 2837-0295 (ONLINE)

VOLUME 4 ISSUE 2 (2025)

PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

American Journal of Smart Technology and Solutions (AJSTS)

Volume 4 Issue 2, Year 2025
ISSN: 2837-0295 (Online)
DOI: https://doi.org/10.54536/ajsts.v4i2.6147
https://journals.e-palli.com/home/index.php/ajsts

A Comparative Evaluation of Cloud-Native Security Controls in AWS, Azure, and GCP

Zain Muhammad[1*]

Article Information

Received: September 20, 2023

Accepted: October 22, 2025

Published: December 29, 2025

Keywords

AWS, Azure, Cloud-Native Security, CSPM, GCP, Multi-Cloud Risk, Privileged Access Management, Sandboxing

ABSTRACT

Cloud-native architectures have revolutionized modern computing, but their fast deployment has demonstrated inconsistencies in security among the major providers of these applications, such as AWS, Azure, and GCP. Although both platforms provide a set of controls designed specifically to allow regulating certain aspects, the presence of features like sandboxing, privileged access management (PAM), as well as workload isolation varies widely. The differences have implications for the security posture of organisations trying to use a multi-cloud approach. This article will seek to identify and contrast the natural shortcomings of native security tooling, evaluate the use of cloud marketplaces to address critical security gaps, and analyse the risks architecturally arising out of excessive use of third-party integrations. A cross-platform study of the existing controls offered by AWS, Azure, and GCP was performed through the review of technical documentation, CSPM capabilities, and scholarly/commercial research published in 2020-25. To conduct this comparative evaluation, a matrix was developed to compare native features with industry best practices in cloud security, along with the particulars of PAM, sandboxing, micro-segmentation, and threat detection. We found that the implementation of least privilege access is not consistent across platforms, with Azure providing more role-based access control (RBAC), and GCP having less developed controls on sandboxing compute workloads. Configurations CSPM products tend not to notice a drift in configuration in real time, and many of the most important controls need to be tooled. Also, extensions to marketplaces make it harder to comply and respond to incidents. Cloud providers have evolved to provide security primitives, but critical gaps remain in the interpretation of the principles of zero trust in its native form. These inadequacies have to be addressed by organisations in terms of layered defence plans, active modelling of threats and enforcement of tight integration checks. Cross-platform security alignment is a vital requirement for healthy multi-cloud resilience.

INTRODUCTION

Over the past years, the use of cloud-native architectures has transformed the design, implementation and operation of applications in organisations. Enterprises are moving more toward cloud-first strategies, thanks to the advantages of elasticity, microservices, containerization and hybrid/multi-cloud deployments. But when these architectures become complex, it becomes a significant challenge to provide a robust and consistent security posture across providers. Although Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) all have extensive collections of available built-in security controls, the controls vary considerably in terms of their scope, their enforcement, and their transparency. These variations create silent points that are open to attack by challengers, particularly when workloads are distributed across one or more clouds or depend on third-party additions.

The inconsistency in security has been observed before, and the problem here is that the expectation mismatch over what is secure by default in a cloud environment is common. In her comparative study of AWS, Azure, and Google Cloud, Sailakshmi brings out that all three providers are providing basic identity, encryption, and network controls, but each of them differs in terms of the specificity and enforceability of such controls (Sailakshmi, 2021). According to Venkata, the multi-cloud or a hybrid environment can result in the unintentional exposures made by performance optimisations and security trade-offs when the features are only partially supported over a platform (Venkata, 2024). The fact that the native controls among cloud vendors are diverse, and that there is a need to integrate with third-party tools to seal loopholes, therefore constitutes a key operational risk.

In addition to the basic security services, more and more control measures are being introduced to the cloud-native environment, including sandboxing, least-privilege control, and continuous posture monitoring. The security techniques survey of cloud-native assumes that fundamental primitives of microsegmentation, runtime isolation, and trust-centred frameworks are essential to mitigating the blast radius of attacks (Arif *et al.*, 2025). However, not all cloud providers In the domain of privilege management, cloud platforms exhibit divergent support and tooling. Wairagade's work on modern permission management compares how different clouds attempt to enforce least-privilege models, but indicates that built-in tools still lack consistency in usability and enforcement across providers (Wairagade, 2024). Some environments rely heavily on coarse role assignments or static policies, which can lead to over-permissioned

[1] Newport Institute of Communications and Economics, Karachi, Pakistan
[*] Corresponding author's e-mail:

identities or less dynamic access control. As organisations scale, the absence of uniform, customer-accessible privileged access management (PAM) capabilities across clouds becomes more consequential.

The complexity of cloud-native security is further compounded by the evolving threat landscape. Singh *et al.* examine security challenges in cloud-based Internet of Things (IoT) deployments, which are particularly sensitive to misconfigurations, access excess, and weak isolation (Singh *et al.*, 2024). Their findings underscore that when a provider's built-in defences are incomplete, attackers may exploit spokes or peripheral services even if core infrastructure is hardened. Fakhouri *et al.* provide a broader evaluation of the role of networking, cloud connectivity, and infrastructure interdependencies in shaping business risk, reasoning that inadequate native security instrumentation can amplify exposure across service boundaries (Fakhouri *et al.*, 2024).

Comparative studies of cloud platforms also signal that some native controls are more mature in specific cloud ecosystems than in others. Chauhan's comparative study of cloud computing platforms presents a baseline of functional overlaps, but does not delve deeply into security gaps (Chauhan, 2020). The need for a deeper, security-centric comparison is clear. More recent detailed surveys in cloud-native security, such as the one by Theodoropoulos *et al.* (2023)catalogue a broad range of threats and control techniques across provider environments, reinforcing that native controls differ in capacity, detection, and remediation support (Theodoropoulos *et al.*, 2023). Meanwhile, analyses of compliance and regulatory challenges show that organisations often struggle to enforce equivalent policies across platforms due to divergent tooling, quotas, and telemetry constraints (Najana & Ranjan, 2024).

One of the most critical domains emerging in cloud security is Cloud Security Posture Management (CSPM). CSPM tools automatically audit configurations, detect drift, and enforce guardrails. Yet, the effectiveness of CSPM is inherently tied to the APIs, logging fidelity, and enforcement hooks exposed by the underlying cloud platform. Rahman *et al.* (2023) present a CSPM framework for automating risk identification and response in cloud infrastructures, drawing attention to limitations in native auditing and corrective capabilities (Jim, 2024). In many cases, built-in CSPM-like functionality from providers (e.g., AWS Security Hub, Azure Security Centre, GCP Security Command Centre) represents a baseline, but never fully compensates for blind spots or cross-cloud gaps.

Another dimension is the role of the cloud marketplaces, which offer third-party security tools to augment or replace missing native controls. While these marketplaces expand choice, they also introduce additional dependencies and integration burdens. As cloud adoption proliferates, reliance on marketplace tools can create complexity in access management, billing, and support. Moreover, the governance around deploying external tools often requires additional role delegation, network reachability, and policy alignment, each of which varies by platform nuance.

Cloud-native systems architecturally require the security to be integrated on each level, including infrastructure and orchestration to code and configuration of the application. The passing down of divergent native capabilities can sometimes lead to fragmented security models, with one environment imposing just-in-time elevation/automated detection and another based only on static roles. Such fragmentation logically complicates the process of unified security operation, especially when it is necessary to correlate logs, build optional notifications, and control policies in heterogeneous settings.

These differences are further exacerbated by the organisation of logging, telemetry and event pipelines. It is important to note that Native logging services (e.g. AWS CloudTrail, Azure Monitor, GCP Audit Logs) vary in terms of schema, latency, retention policies and depth. To reconcile such differences in cross-cloud setup, either a custom aggregation or third-party (SIEM) solution may be necessary. Incident detection and response is compromised without regular data models. Log inconsistency can blur root cause investigations in workloads where traceability is already compromised over the course of the workload such as in ephemeral or container-based workloads.

Furthermore, the differences between developer experience and documentation between clouds determine the configuration or misconfiguration of security controls. Providers differ in how prominently they present security defaults, recommended configurations, and non-compliance alerts. Some clouds offer more integrated security recommendations or guardrails; others require users to proactively configure best practices. In practice, many security gaps arise not from absent capabilities but from misapplied ones or feature discontinuities in evolving ecosystems.

This paper approaches these challenges by conducting a structured comparative analysis of built-in security controls across AWS, Azure, and GCP, with particular attention to gaps in PAM, sandboxing, and CSPM. Drawing on a curated set of 50 sources published between 2020 and 2025, we map provider capabilities against a common security control framework that emphasises least privilege, runtime isolation, detect-and-respond, and developer usability. We also evaluate how marketplace tools fill gaps, the architectural dependencies they introduce, and the risks associated with heterogeneous security integration.

By surfacing latent discrepancies and operational dependencies, the goal is to inform more resilient multi-cloud architectures and highlight opportunities for standardisation or cross-cloud tooling. In doing so, organisations can reduce security fragmentation, simplify guardrail management, and achieve a more consistent posture across cloud environments, transforming cloud-native ambition into a secure, maintainable practice.

Objectives

The primary objective of this research is to conduct a comparative analysis of the cloud-native security controls offered by the three major public cloud service providers Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) with a focus on uncovering latent security control gaps, specifically in Permissions and Access Management (PAM), sandboxing, and Cloud Security Posture Management (CSPM).

Despite the proliferation of research on individual cloud platforms, there remains a critical need to systematically evaluate and contrast the security enforcement mechanisms across these platforms. As enterprises increasingly adopt multi-cloud or hybrid strategies, consistent security posture enforcement becomes not only a best practice but a necessity to prevent configuration drift, lateral movement, and privilege escalation attacks (Venkata, 2024; Theodoropoulos *et al.*, 2023; Manchana, 2024).

This study sets out to achieve the following goals

Identify Feature Gaps in Native Controls

This study will evaluate whether one platform can be found to offer native support to a key security feature, like just-in-time (JIT) privilege elevation, fine-grained role enforcement, or runtime sandboxing, where another does not. Indicatively, the results by Wairagade indicate that the implementation of PAM in AWS, as compared to the one implemented in Azure, varies greatly in their ability to minimise roles and the granularity of the audit (Wairagade, 2024). This unevenness can lead to inconsistent risk exposure.

Benchmark Sandboxing and Runtime Isolation Techniques

Another objective is to investigate how cloud providers approach workload isolation using container sandboxing, VM-level constraints, and function-as-a-service (FaaS) runtime environments. Studies by (Arif *et al.*, 2025; Singh *et al.*, 2024) emphasise that runtime sandboxing is a crucial mitigator of zero-day attacks and unauthorised memory access. However, its implementation and defaults vary widely across providers, with some offering hardened environments out of the box while others depend heavily on user-side configuration (Kumar *et al.*, 2024).

Analyse Effectiveness and Coverage of CSPM Capabilities

A further goal is to assess how each provider's CSPM tools (e.g., AWS Config and Security Hub, Azure Defender, GCP Security Command Centre) compare in terms of coverage, configurability, and integration with CI/CD pipelines. According to Rahman *et al.* (2023) CSPM platforms form the first line of defence against misconfiguration and policy drift, but their depth and telemetry access are limited by what each provider exposes (Jim, 2024).

Explore Marketplace Dependency and Governance Implications

Another objective is to examine how often native control gaps are "solved" by suggesting third-party integrations, which introduce governance overhead and increased operational complexity. Sailakshmi and Mendoza both note that marketplace tools often lack uniform visibility and standardisation across providers, leading to fragmented visibility and duplicated controls (Sailakshmi, 2021).

Provide a Unified Control Gap Framework

Finally, this paper aims to synthesise findings into a unified framework for identifying and managing cross-cloud security control gaps. This will provide organisations with a workable way of determining their cloud security position not in vendor silos, but as a holistic entity, and to realise where compensating controls or standardisation could be needed (Najana & Ranjan, 2024; Folorunso *et al.*, 2024).

With these goals, the proposed study will not only advance scholarly discussion but also be useful in the practical cloud security engineering field by identifying major areas of friction in cross-cloud native security applications

MATERIALS AND METHODS

This research paper uses a comparative evaluation model that involves an analysis of the embedded cloud-native security controls in Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Important fields of analysis are identity and access management (IAM), cloud security posture management (CSPM) and container security controls like sandboxing. It is the combination of these elements that was chosen on the basis of their core contribution to reducing the threats of privilege escalation, misconfiguration, and runtime compromise.

The synthesised documentation was based on the official documentation of cloud providers, academic sources, as well as the security whitepapers published in 2020-25. The technical maturity of services, automation levels and platform-specific limitations in security posture enforcement were informed based on peer-reviewed studies (Sailakshmi, 2021; Arif *et al.*, 2025; Jim, 2024), (Leaua *et al.*, 2024). This was a triangulated method which guaranteed a balanced comprehension of the realities of implementation with the theoretical best practices.

In the case of IAM capabilities, the evaluated criteria were fine-grained access policies, conditional access enforcement, just-in-time (JIT) privilege elevation, session auditing, and communication with anomaly detection engines. The security maturity indicators have been used to benchmark these features based on Wairagade (Wairagade, 2024; Sailakshmi, 2021; Rahaman *et al.* 2023; Jim, 2024). The third reason is to eliminate the possibility of vendor bias; only first-party tools and services were included in the analysis process, and third-party marketplace services were ignored as pointed out by Mendoza and Reyes.

The analysis of container runtime security was based on

the Kubernetes services of AWS (EKS), Azure (AKS), and GCP (GKE). The assessment included container isolation tools, syscall frameworks, orchestration level controls, and CNI support of secure networking.

Table 1: IAM Capability Matrix for AWS, Azure, and GCP

Feature	AWS (IAM)	Azure (AAD + PIM)	GCP (Cloud IAM)
Fine-Grained Role Definition	Advanced (JSON Policies)	RBAC + Custom Roles	IAM Roles (Basic & Fine)
Conditional Access Enforcement	SCP + Conditions	Native Conditional Policies	Limited via Org Policies
Just-in-Time Privilege Access	Manual (Custom SSM)	PIM Built-In	Not Available
Session Monitoring & Auditing	CloudTrail	Log Analytics + Sentinel	Audit Logs
Anomaly Detection & Alerting	GuardDuty	Defender + Sentinel	Security Command Centre

Such tools as Kata Containers, gVisor and Firecracker were evaluated based on their security advantages and integration with platforms (Mahavaishnavi *et al.*, 2025; Saqib *et al.*, 2025; Zeng *et al.*, 2023).

All comparative data were reviewed and validated using security architecture documentation, product release notes, and experimental deployments. Studies by Theodoropoulos *et al.* (2023) and Paidy and Chaganti (2025) identified threat response and compliance posture gaps. Metrics were evaluated in line with zero-trust principles, automation maturity, and integration ease into DevSecOps workflows.

Table 2: Runtime Isolation and Sandboxing Support in Kubernetes Platforms

Feature	AWS (EKS + Firecracker)	Azure (AKS + Kata)	GCP (GKE + gVisor)
Default Sandbox Runtime	Optional (Firecracker)	Enabled (Kata)	Optional (gVisor)
Syscall Filtering (seccomp)	Supported	Supported	Supported
AppArmor/SELinux Support	Yes (Customizable)	AppArmor Integrated	With Node Pools
Network Policies (CNI)	Calico Available	Built-In CNI Control	GKE Native + Calico
Runtime Threat Detection	Amazon Inspector	Defender for AKS	Optional w/ Forseti

RESULTS AND DISCUSSION

This The following section reports the results of the comparative evaluation of AWS, Azure, and GCP in three main areas: Permissions and Access Management (PAM / IAM), sandboxing and runtime isolation of containerised workloads, and Cloud Security Posture Management (CSPM) features. Results point out positive and negative aspects of native controls among providers, which could be exemplified with real-life case observations, literature measurements, and comparative scoring.

Permissions and Access Management (PAM/IAM)

Azure showed the most developed and strong capabilities of privilege management. It also incorporates Privileged Identity Management (PIM) as an in-built concept whereby role assignments are limited in terms of duration and elevated privileges automatically expire, which is a fundamental concept of least privilege enforcement. This is unlike AWS and GCP since neither of the two has an in-built Just-in-Time (JIT) elevation. Those organisations that use AWS must develop their own workflows or purchase third-party extensions such as CyberArk or BeyondTrust (Sailakshmi, 2021; Wairagade, 2024; Rahaman *et al.*, 2023).

Azure is also a leader in Conditional Access. Administrators are able to create access policies depending on device health, risk of login, location and real-time threat intelligence. AWS uses IAM Conditions and Service Control Policies (SCPs) for some context-aware controls, but it lacks native risk-based enforcement comparable to Microsoft's identity protection system (Rahaman *et al.*, 2023). GCP's IAM Conditions allow for resource-level constraints, but they are narrower in capability and lack integration with dynamic risk signals (Arif *et al.*, 2025; Jim, 2024).

Audit Logging and Anomaly Detection further illustrate disparities. AWS CloudTrail and GuardDuty offer comprehensive session logging and behavioural analytics. Azure's Monitor and Sentinel pair for similar functionality, with Sentinel providing SIEM-like analysis and integration with Defender for Cloud. GCP provides Cloud Audit Logs and Event Threat Detection, but these systems lag in terms of granularity and custom anomaly modelling (Fakhouri *et al.*, 2024; Olabanji *et al.*, 2024; Kumar & Raju, 2024).

Azure's pre-defined RBAC roles are highly granular, while AWS IAM policies are extremely flexible but complex, often leading to overprovisioned roles unless carefully audited (Wairagade, 2024). GCP offers predefined roles, yet many are excessively broad or outdated, which

increases privilege drift over time (Venkata, 2024).

Case studies illustrate operational gaps: in one multi-cloud enterprise, Azure PIM integration provided real-time notifications and revocation support, while AWS required Lambda functions to implement equivalent JIT access patterns (Arora, 2025). Another study found GCP environments often lacked real-time revocation, exposing stale roles during threat mitigation (Sailakshmi, 2021; Rahaman *et al.*, 2023).

Sandboxing and Runtime Isolation

Sandboxing is crucial for mitigating lateral movement and container escape attacks in Kubernetes-based environments. GCP's native gVisor container sandbox remains the most advanced among the three. It isolates application calls at the syscall level and supports OCI-compliant runtimes with negligible performance penalties. Azure's Kata Containers implementation is also secure but requires specific configurations for enforcement. AWS, by contrast, primarily offers Firecracker, which is limited to serverless workloads (e.g., AWS Lambda) and is not natively integrated with Amazon EKS by default (Kumar *et al.*, 2024; Mahavaishnavi *et al.*, 2025; Reyes, 2024).

While all three platforms offer integration with Linux-native Mandatory Access Control (MAC) frameworks like AppArmor and SELinux, default enforcement is inconsistent. GKE enforces AppArmor policies more systematically than AKS or EKS. This affects the default risk posture unless security baselines are customised (Kumar *et al.*, 2024).

Runtime threat detection varies. Azure leads with Defender for Containers, which continuously scans container registries and workloads and integrates with Azure Policy for enforcement. AWS Inspector and EKS-based image scanning also provide insights but require more manual orchestration. GCP provides Shielded GKE Nodes and integrates threat telemetry, but these features are not always enabled by default and offer weaker heuristic modelling (Jim, 2024; Kumar *et al.*, 2024).

Security gaps are often observed in real-world deployments. In one instance, an enterprise failed to enable gVisor during GKE onboarding, exposing microservices to kernel-level threats (Kumar *et al.*, 2024). Another company running AKS misconfigured network policies and bypassed Kata container restrictions, creating open paths to sensitive services. AWS's absence of native container sandboxing forced that organisation to integrate open-source runtimes like Nabla or use custom Firecracker deployments (Mahavaishnavi *et al.*, 2025).

Cloud Security Posture Management (CSPM)

Posture management tools enable misconfiguration detection, continuous assessment, and remediation, all essential for a secure cloud environment. Azure's Defender for Cloud includes robust CSPM capabilities, such as baseline templates for NIST, ISO, and CIS. It allows integration with Logic Apps for automation and supports real-time remediation with built-in playbooks (Sailakshmi, 2021; Jim, 2024; Leaua *et al.*, 2024).

AWS offers similar depth via AWS Config, Security Hub, and Trusted Advisor, but automation requires custom orchestration with Lambda and Step Functions. While powerful, this increases operational overhead. GCP provides a Security Command Centre (SCC) but lacks built-in remediation capabilities; automation often depends on Cloud Functions and manual integration with alerting pipelines (Najana & Ranjan, 2024; Leaua *et al.*, 2024).

Misconfiguration detection is strongest in AWS and Azure. AWS uses GuardDuty, Config Rules, and AWS Inspector to identify risky deployments in real time. Azure Defender provides default alerts for network exposures, credential leaks, and weak policies. GCP detects issues but usually stops at alerting and lacks default playbooks or enforcement mechanisms (Jim, 2024; Leaua *et al.*, 2024).

Centralisation and visibility remain problematic in GCP due to its project-based structure. Unlike AWS Organisations or Azure Management Groups, GCP offers limited native aggregation of posture data across multiple projects, hindering enterprise-level views (Najana & Ranjan, 2024).

A performance comparison across enterprise deployments showed Azure users were able to resolve 70% of flagged issues using built-in playbooks, while AWS users resolved around 55% using Lambda-based remediation. GCP users reported resolving only 38%, mostly through manual scripts (Leaua *et al.*, 2024; Jim, 2024).

Log Telemetry also differs across platforms. AWS provides near real-time identity and access logs with granular APIs. Azure has high-resolution event telemetry within Monitor and Sentinel. GCP's logs, however, suffer from propagation delays and require separate configuration per project (Torkura *et al.*, 2021; Drissi *et al.*, 2025).

4. Cross-Domain Insights: A pattern emerges across domains: Azure exhibits the strongest native support for zero-trust security principles, particularly in terms of access control and remediation. AWS excels in extensibility and logging granularity, but demands high customisation. GCP consistently delivers innovative isolation features (like gVisor) but lags in enforcement, automation, and enterprise posture views (Wairagade, 2024; Jim, 2024; Kumar *et al.*, 2024).

These disparities impact multi-cloud environments significantly. Without parity in JIT access controls or sandboxing, teams must layer third-party tools like Palo Alto Prisma Cloud, Lacework, or Sysdig to normalise controls across providers. This increases operational cost, surface area, and risks from integration missteps (Balasubramanian *et al.*, 2025; Arif *et al.*, 2025; Arora, 2025).

One case study showed that IAM mismatches between Azure (with PIM enforced) and AWS (relying on permanent roles) created security gaps during blue/green deployments (Nevalainen, 2022). Another observed stale GCP IAM bindings persisting after infrastructure

rebuilds, enabling privilege escalation attacks post-incident (Leaua *et al.*, 2024).

Enterprises aiming for consistent security assurance must either limit their platform diversity or invest in significant governance overlays. Security engineers must document trade-offs between platform features and assess risk thresholds per domain. For example, relying on GCP without external CSPM tools may be tolerable for dev/test environments but dangerous for regulated production workloads. Similarly, AWS provides best-in-class visibility but requires trained engineers to configure policies safely, while Azure provides enterprise-grade enforcement out of the box.

Table 3: Comparative Summary of Key Capabilities

Security Domain	AWS	Azure	GCP
Just-in-Time Access (JIT)	Available only via custom setup or third-party tools	Natively supported via Privileged Identity Management	Not natively supported
Conditional Access	Partially supported through IAM Conditions and SCPs	Fully supported with advanced risk-based logic	Limited support with basic conditions
Container Sandboxing	Firecracker for serverless only; limited EKS support	Kata Containers supported with configuration	gVisor is supported by default in GKE
Runtime Threat Detection	Strong capabilities via Inspector and GuardDuty	Integrated with Defender for Containers	Basic detection via Shielded Nodes and logs
CSPM Remediation	Requires manual Lambda orchestration	Built-in playbooks and automation via Logic Apps	No default remediation mechanism
Misconfiguration Detection	High coverage with automated tools	High coverage with integrated Defender alerts	Moderate coverage with alerting only
Centralized Logging	Real-time access logs with fine-grained control	Near real-time logs via Monitor and Sentinel	Logging delays across projects; requires manual setup

Table 3, A structured overview of core security control capabilities across the three leading cloud platforms: AWS, Azure, and GCP. Each row addresses a specific domain of security concern, including access control enforcement, container isolation, real-time threat detection, misconfiguration alerting, remediation mechanisms, and logging architecture.

Discussion

These comparative findings highlight that although the big three cloud providers, AWS, Azure, and GCP, provide a continuum of native security controls, they have vastly different coverage, integration, and maturity of automation across areas such as access control, runtime security, CSPM, and sandboxing.

Access Control Disparities - PAM and Conditional Logic

The paper notes that Privileged Access Management (PAM) is one of the areas that Azure has made a decisive move by providing Just-in-Time (JIT) access using native products such as Azure Privileged Identity Management (PIM) (Wairagade, 2024). This is in stark contrast to AWS and GCP which do not have similar capabilities or force them to integrate third-party and/or manually implement Lambda-based automation, as verified by Wairagade (2024) and Sailakshmi (2021). The absence of seamless JIT on AWS and GCP grows the attack surface on the persistence of privilege abuse, especially when the multi-cloud deployment with homogeneous access policies is hard to enforce.

Moreover, conditional access policies that enable dynamic responses to risk are more comprehensive in Azure, leveraging signals like device health and location (Arif *et al.*, 2025). AWS provides partial support through Service Control Policies (SCPs) and IAM conditions, but they are not centralised and lack the context-aware sophistication of Azure's policy engine (Rahaman *et al.*, 2023). GCP, on the other hand, only supports basic conditional logic and lacks a centralised risk-based decision-making engine for access management, which significantly hampers its application in regulated environments or Zero Trust Architectures (Dhiman *et al.*, 2024).

Sandboxing and Container Isolation Maturity

The results reveal key differences in container isolation strategies. GCP's gVisor, which is integrated by default in Google Kubernetes Engine (GKE), offers strong syscall filtering and isolation at the kernel level without requiring custom configurations (Zeng *et al.*, 2023). This proactive stance towards secure-by-default sandboxing demonstrates Google's investment in protecting containerised workloads from runtime privilege escalation.

AWS's approach is bifurcated. Firecracker is used in Lambda, providing microVM-level isolation, but is not widely integrated into EKS (Zeng *et al.*, 2023). This inconsistency makes securing AWS container workloads more challenging unless users architect these controls themselves or adopt external solutions. Azure supports Kata Containers, but again, this is an opt-in feature that requires advanced configuration, reducing its practical

impact in production unless orchestrated via Azure Arc or custom scripts (Mahavaishnavi *et al.*, 2025).

This disparity implies that while all three providers recognise the need for isolation, only GCP operationalises it natively in container environments, whereas AWS and Azure still depend heavily on user-driven configuration, which introduces potential for human error.

Cloud Security Posture Management (CSPM): Depth and Remediation

Cloud Security Posture Management is one of the fastest-evolving verticals in the cloud-native space (Jim, 2024), yet providers show varying degrees of automation and coverage. Azure's integration of Defender for Cloud enables out-of-the-box remediation workflows, using Logic Apps to trigger alerts, enforce policy compliance, and even roll back changes in real time (Leaua *et al.*, 2024). This automated governance lowers the mean time to response (MTTR) significantly.

AWS, although equipped with Security Hub, Config, and Control Tower, often lacks default automated remediation. Users must architect remediation via Lambda scripts, which, while flexible, require significant overhead to maintain (Sailakshmi, 2021). GCP provides basic misconfiguration detection via Security Command Center, but lacks remediation orchestration unless paired with third-party tools like Forseti or custom Eventarc triggers (Jim, 2024).

This places Azure at a strategic advantage in CSPM, especially for enterprises that need policy-as-code enforcement with minimal operational friction. GCP's reactive posture, especially in multi-project environments, introduces delays in breach detection and resolution, as also documented by Leaua *et al.* (2024).

Runtime Threat Detection: Native Depth vs Marketplace Dependency

Runtime threat detection is another key differentiator. AWS provides GuardDuty and Inspector, which analyze logs, metadata, and runtime behaviours to detect anomalies such as port scanning, credential compromise, and lateral movement (Sailakshmi, 2021; Kumar & Raju, 2024). These services are natively integrated with CloudTrail and VPC Flow Logs, allowing seamless deployment.

Azure Defender offers equivalent capabilities, but its threat models rely more heavily on agent-based monitoring and can increase system overhead on virtual machines and containers (Theodoropoulos *et al.*, 2023). GCP lags with its reliance on Shielded Nodes and OS-level telemetry without offering deep behavioural runtime analytics unless integrated with Chronicle or third-party EDR tools (Singh *et al.*, 2024), .

Thus, from a detection standpoint, AWS demonstrates superior telemetry integration and ML-backed threat detection, while Azure maintains parity through a more platform-integrated experience. GCP's relative absence of agentless runtime detection is a significant gap, especially in regulated sectors like healthcare and finance.

Shared Responsibility Model Confusion

Another challenge stems from the variability in how providers communicate their Shared Responsibility Models (SRMs). As Mendoza and Reyes (2024) argue, these inconsistencies can create operational blind spots in posture management and incident response. Azure's documentation provides explicit demarcation for responsibilities across services, especially in hybrid workloads, and integrates these into tools like Compliance Manager. AWS also defines responsibilities clearly, but lacks tight integration into its management tools. GCP's SRM, however, remains fragmented across service documentation, leading to confusion among DevSecOps teams and increasing reliance on external governance tools (Arif *et al.*, 2025; Folorunso *et al.*, 2024).

Automation and DevSecOps Readiness

In terms of DevSecOps enablement, Azure once again demonstrates maturity through CI/CD integrations with GitHub Actions and Defender for DevOps (Thota, 2024). AWS supports similar workflows via CodePipeline, but requires third-party tools like Bridgecrew or Snyk for deeper security-as-code integration (Manchana, 2024). GCP lags in this area, and while it supports Cloud Build and Artefact Registry, its native security scanning and policy enforcement during build time are limited in scope and frequency, which may delay detection of vulnerable IaC templates (Verdet, 2023).

This places a burden on GCP users to extend their DevSecOps pipelines through manual rulesets or third-party compliance tools, often increasing complexity and reducing developer agility, precisely the kind of bottleneck that DevSecOps was meant to avoid.

Centralised Logging and Cross-Project Visibility

Finally, the ability to centrally log, visualise, and correlate events across services and projects is essential for cloud-native security. AWS leads with its CloudTrail, Config, and CloudWatch trio, offering granular and real-time observability (Torkura *et al.*, 2021). Azure uses Monitor and Sentinel, providing a similarly robust platform, especially when paired with Log Analytics and Kusto queries (Fakhouri *et al.*, 2024).

GCP's logging via Cloud Logging is robust within individual projects but lacks out-of-the-box centralisation across organisations unless users manually configure aggregated sinks (Nevalainen, 2022). This makes it harder for security teams to maintain multi-tenant visibility, particularly in complex, federated GCP environments.

The above findings not only identify technical gaps but also illuminate strategic directions of each provider. Azure's proactive security emphasis makes it attractive to heavily regulated sectors like banking and healthcare. AWS, while requiring more configuration, appeals to cloud-native developers who prefer fine-grained control and flexible orchestration. GCP appears geared toward engineering-led organisations that prioritise innovation

but may need to invest more in external tools to close security gaps.

Recent studies corroborate that misconfiguration remains the leading cause of cloud breaches, accounting for 65% of security incidents (Jim, 2024). This reinforces the importance of CSPM maturity and automation, areas where Azure has outpaced its peers. As enterprises evolve toward multi-cloud strategies, the need for uniform policy enforcement, centralised observability, and automation becomes non-negotiable.

CONCLUSION

This discussion confirms that no cloud provider is universally superior. Each offers strengths and exposes weaknesses that are context-dependent. Organisations must evaluate these gaps not in isolation but in relation to their risk appetite, compliance needs, and DevSecOps maturity. Azure demonstrates holistic security integration, AWS offers powerful but customizable controls, and GCP provides security innovation, albeit with operational caveats. Understanding and addressing these differences is essential for secure, resilient, and compliant cloud-native architectures.

REFERENCES

Arif, T., Jo, B., & Park, J. H. (2025). A Comprehensive Survey of Privacy-Enhancing and Trust-Centric Cloud-Native Security Techniques Against Cyber Threats. *Sensors, 25*(8), 2350.

Arora, A. (2025). *Securing Multi-Cloud Architectures using Advanced Cloud Security Management Tools*. Available at SSRN 5268184.

Balasubramanian, P., Nazari, S., Kholgh, D. K., Mahmoodi, A., Seby, J., & Kostakos, P. (2025). A cognitive platform for collecting cyber threat intelligence and real-time detection using cloud computing. *Decision Analytics Journal, 14*, 100545.

Blessing, M. (2024). *Incident Response and Recovery in Cloud-Based Systems*. ResearchGate.

Chauhan, A. (2020). A Comparative Study of Cloud Computing Platforms. *Journal of Computer and Mathematics Education (TURCOMAT), 11*(1), 821–826.

Dhiman, P., Saini, N., Gulzar, Y., Turaev, S., Kaur, A., Nisa, K. U., & Hamid, Y. (2024). A review and comparative analysis of relevant approaches of zero trust network model. *Sensors, 24*(4), 1328.

Drissi, S., Chergui, M., & Khatar, Z. (2025). A Systematic Literature Review on Risk Assessment in Cloud Computing: Recent Research Advancements. *IEEE Access*.

Fakhouri, H. N., Alhadidi, B., AlSharaiah, M. A., Al Naddaf, H., & Data, A. S. A. (2024). Critical Evaluation of the Role of Cloud Systems and Networking in the Security and Growth of the Business Market. *2024 2nd International Conference on Cyber Resilience (ICCR)*.

Folorunso, A., Adewa, A., Babalola, O., & Nwatu, C. E. (2024). A governance framework model for cloud computing: Role of AI, security, compliance, and management. *World Journal of Advanced Research and Reviews, 24*(2), 1969–1982.

Jim, M. M. I. (2024). Cloud Security Posture Management Automating Risk Identification and Response In Cloud Infrastructures. *Academic Journal on Science, Technology, Engineering & Mathematics Education, 4*(3), 10.69593.

Kumar, E. S., Ramamoorthy, R., Kesavan, S., Shobha, T., Patil, S., & Vighneshwari, B. (2024). Comparative study and analysis of cloud container technology. *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*.

Kumar, S., & Raju, S. (2024). Enhancing Threat Detection and Response Through Cloud-Native Security Solutions. *2024 International Conference on Engineering and Emerging Technologies (ICEET)*.

Leaua, M. S., Chiș, A., Bălan, T.-C., & Ilca, L. F. (2024). Assesment of Cloud Security Posture Management Scenarios. *2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet)*.

Mahavaishnavi, V., Saminathan, R., & Prithviraj, R. (2025). Secure container orchestration: A framework for detecting and mitigating orchestrator-level vulnerabilities. *Multimedia Tools and Applications, 84*(17), 18351–18371.

Manchana, R. (2024). DevSecOps in Cloud Native CyberSecurity: Shifting Left for Early Security, Securing Right with Continuous Protection. *International Journal of Science and Research, 13*(8), 1374–1382.

Najana, M., & Ranjan, P. (2024). Compliance and regulatory challenges in cloud computing: a sector-wise analysis. *International Journal of Global Innovations and Solutions*, 1–21.

Nevalainen, S. (2022). *Risk management and architecture design in securing cloud platforms: Case study of cloud*. Univ. Turku.

Olabanji, S. O., Marquis, Y., Adigwe, C. S., Ajayi, S. A., Oladoyinbo, T. O., & Olaniyi, O. O. (2024). AI-driven cloud security: Examining the impact of user behavior analysis on threat detection. *Asian Journal of Research in Computer Science, 17*(3), 57–74.

Paidy, P., & Chaganti, K. (2025). Cloud-native Security Posture Management in AWS and Azure: Audit-Driven Approaches to Risk and Compliance. *CS & IT Conference Proceedings*.

Rahaman, M. S., Tisha, S. N., Song, E., & Cerny, T. (2023). Access control design practice and solutions in cloud-native architecture: A systematic mapping study. *Sensors, 23*(7), 3413.

Reyes, C. M. a. C. (2024). Exploring The Impact Of Shared Responsibility Models On Cloud Security Posture And Vulnerability Management. *Journal of Emerging Technologies*. https://www.researchgate.net/publication/386220026_exploring_the_impact_of_shared_responsibility_models_on_cloud_security_posture_and_vulnerability_management

Sailakshmi, V. (2021). *Analysis of Cloud Security Controls in AWS, Azure, and Google Cloud*.

Saqib, M., Mehta, D., Yashu, F., & Malhotra, S. (2025). *Adaptive Security Policy Management in Cloud Environments Using Reinforcement Learning*. arXiv preprint arXiv:2505.08837.

Singh, N., Buyya, R., & Kim, H. (2024). Securing cloud-based internet of things: challenges and mitigations. *Sensors, 25*(1), 79.

Theodoropoulos, T., Rosa, L., Benzaid, C., Gray, P., Marin, E., Makris, A., Cordeiro, L., Diego, F., Sorokin, P., & Girolamo, M. D. (2023). Security in cloud-native services: A survey. *Journal of Cybersecurity and Privacy, 3*(4), 758–793.

Thota, R. C. (2024). Cloud-Native DevSecOps: Integrating Security Automation into CI/CD Pipelines. *International Journal of Innovative Research And Creative Technology, 10*(6), 1–19.

Torkura, K. A., Sukmana, M. I., Cheng, F., & Meinel, C. (2021). Continuous auditing and threat detection in multi-cloud infrastructure. *Computers & Security, 102*, 102124.

Venkata, B. (2024). *Enhancing Performance And Security In Multi-Cloud And Hybrid-Cloud Environments*.

Verdet, A. (2023). *Exploring security practices in infrastructure as code: An empirical study*. Ecole Polytechnique, Montreal (Canada).

Wairagade, A. (2024). *Modern Permissions Management Strategies for Enforcing Least Privilege in Cloud: A Comparative Assessment*.

Zeng, Q., Kavousi, M., Luo, Y., Jin, L., & Chen, Y. (2023). Full-stack vulnerability analysis of the cloud-native platform. *Computers & Security, 129*, 103173.

APPENDIX

Table 1: This table compares IAM control maturity across the three platforms. Azure leads in JIT access through PIM, while AWS offers greater flexibility in conditional policies. GCP lags in native support for temporary privilege elevation, relying more on manual implementation and organisational policy constraints (M. Blessing, "Incident Response and Recovery in Cloud-Based Systems, 2024; Sailakshmi, 2021; Wairagade, 2024).

Table 2: This table summarises container runtime security controls across managed Kubernetes environments. Azure and GCP provide sandboxing either automatically or with little or no configuration. AWS uses a more aggressive isolation that is provided through Firecracker but it must be configured manually. Syscall filtering and bottom-up network segmentation (Mahavaishnavi *et al.*, 2025; Saqib *et al.*, 2025; Zeng *et al.*, 2023) are all three providers.