

A Proactive Approach to Secure Multi-Cloud through Compliance and Governance by Design

*Neha Chandrima
**Dr. Rajesh Kumar Nigam

ABSTRACT

Multi-cloud adoption enables organizations to use the capabilities of different cloud service providers for flexibility, resilience, cost optimization, and service availability. At the same time, heterogeneous cloud environments create difficulties in maintaining consistent security controls, governance, audit readiness, and regulatory compliance. This paper presents a structured research framework for a proactive approach to securing multi-cloud environments through Compliance and Governance by Design. The proposed approach embeds compliance controls into infrastructure provisioning and DevSecOps pipelines rather than relying primarily on post-deployment audits. Policy as Code (PaC) is used as the central mechanism for expressing and enforcing security and compliance requirements in a repeatable manner. The proposed framework targets AWS, Azure, and Google Cloud Platform and considers controls such as encryption, least-privilege identity and access management, logging, monitoring, and configuration validation. A methodology is proposed comprising requirement analysis, framework design, CI/CD integration, multi-cloud deployment, and evaluation. Candidate evaluation measures include compliance drift detection rate, mean time to compliance, audit readiness, and deployment latency overhead. The paper does not claim completed experimental results; instead, it converts the supplied doctoral synopsis into a publication-oriented research framework and identifies the research gap, implementation direction, evaluation plan, and expected contribution.

Keywords:- Multi-Cloud Security, Compliance by Design, Governance by Design, Policy as Code, DevSecOps, Infrastructure as Code, Cloud Compliance, Compliance Drift, AWS, Azure, GCP.

*Neha Chandrima, Research Scholar, Department of Computer Science and Engineering, Oriental University, Indore (M.P.), Chandrima2006@gmail.com.

**Dr. Rajesh Kumar Nigam, Associate Professor, Department of Computer Science and Engineering, Oriental University Indore (M.P.).

I. INTRODUCTION

Cloud computing has become an important foundation for modern digital services because it provides scalable, flexible, and cost-effective computing, storage, networking, and application delivery capabilities. However, moving workloads and data to cloud platforms introduces security responsibilities that differ from those of conventional infrastructure. The security of cloud resources is influenced by the shared responsibility model, in which the cloud provider and the customer have different responsibilities for protecting infrastructure, applications, data, identities, and configurations.

The distributed and multi-tenant characteristics of cloud platforms increase the importance of systematic security controls. Threats can arise from malicious attacks, data breaches, insider threats, insecure interfaces, inadequate access control, and configuration errors. In a multi-cloud environment, these challenges are compounded because organizations may use different security services, policy languages, configuration models, identity mechanisms, and compliance capabilities across providers.

Organizations increasingly adopt AWS, Azure, GCP, and other cloud services to obtain flexibility and resilience. The supplied doctoral synopsis identifies that this heterogeneity creates significant governance and compliance challenges. Traditional compliance approaches based heavily on manual audits and post-deployment checks may not provide sufficient protection against rapidly changing configurations. Misconfiguration and compliance drift can occur between audits, creating operational and regulatory risk.

A proactive approach therefore requires compliance controls to be incorporated into the design and delivery process. Compliance and Governance by Design treats compliance as an engineering requirement that should be evaluated before and during resource deployment. Policy as Code provides a mechanism for expressing rules in machine-readable form and enforcing them consistently. When these controls are integrated into CI/CD pipelines, infrastructure can be validated before deployment and continuously evaluated after deployment.

The proposed research addresses this need by developing and evaluating a compliance-by-design framework for heterogeneous multi-cloud environments. The framework combines infrastructure as code, policy enforcement, DevSecOps automation, cloud-native compliance mechanisms, and centralized evaluation. Its intended outcome is a reusable approach capable of reducing compliance drift, improving audit readiness, and strengthening security governance across multiple providers.

II. LITERATURE REVIEW

Research on cloud security has developed from foundational cloud-computing and shared-responsibility concepts toward automated governance, continuous compliance, and infrastructure hardening. The supplied synopsis reviews work covering encryption, identity and access management, network security, monitoring, policy enforcement, compliance, and cloud infrastructure hardening.

Armbrust et al. discussed cloud computing and the division of responsibilities between cloud providers and consumers. Zissis and Lekkas emphasized cloud security issues and the importance of security mechanisms for protecting cloud resources. The synopsis also identifies research addressing encryption, access control, monitoring, network segmentation, key management, automated incident response, and compliance governance.

The literature summarized in the synopsis indicates that provider-specific security tools can support compliance but may not provide uniform enforcement across heterogeneous cloud environments. Work on cloud governance has highlighted policy-based auditing and remediation, while studies of security configuration have identified the continuing challenge of maintaining secure configurations at scale.

The selected base paper in the synopsis, by Kousalya A. and Baik Nam-kyun (2023), proposes improved RSA-based RBAC with XACML for cloud security. The synopsis identifies this work as a Scopus-indexed base paper and notes its relevance to identity management, data protection, and compliance. The proposed doctoral work extends the research direction beyond access control toward integrated, multi-cloud compliance and governance.

The literature also identifies the value of continuous monitoring, policy enforcement, IAM, MFA, network security, encryption, and security information and event management. However, the synopsis identifies a gap in obtaining consistent, proactive enforcement across heterogeneous cloud platforms. The proposed work therefore focuses on integrating policy validation into provisioning and CI/CD processes.

Table 1: Selected Literature and Research Relevance

Author/Year	Area / Method	Major Contribution	Identified Limitation / Gap
Armbrust et al. (2010)	Cloud computing	Discussed foundations and cloud adoption.	Does not provide a complete multi-cloud compliance-by-design framework.
Zissis & Lekkas (2012)	Cloud security	Addressed security issues and protection	Focus is broad cloud security rather than automated multi-cloud

		mechanisms.	governance.
Anderson et al. (2016)	Data security	Emphasized encryption and cloud data protection.	Does not integrate controls into a unified multi-cloud DevSecOps pipeline.
Harris et al. (2020)	IAM	Emphasized roles, authentication and access control.	Provider-specific implementation does not solve heterogeneous governance.
Chen & Zhao (2019)	Security monitoring	Reviewed monitoring and detection approaches.	Focuses on monitoring rather than infrastructure compliance hardening.
Liu et al. (2019)	Compliance and governance	Discussed cloud policy and governance.	Limited coverage of cross-provider enforcement.
Kousalya & Baik Nam-kyun (2023)	RSA/RBAC/XACML	Proposed improved access-control and encryption approach.	Research is primarily focused on access control rather than complete multi-cloud governance.
Batchu (2025)	Cloud infrastructure fortification	Discussed advanced cloud protection strategies.	Does not establish the proposed integrated compliance-by-design pipeline.

III. RESEARCH GAP

The literature reviewed in the supplied synopsis demonstrates substantial progress in cloud security, but it also indicates a continuing need for proactive and consistent compliance enforcement across heterogeneous environments. Provider-specific tools may enforce policies within individual platforms, yet organizations operating across AWS, Azure, and GCP need a common governance approach.

A key gap is the separation between compliance requirements and the software delivery process. When compliance is assessed mainly after deployment, a configuration can reach a production environment before a violation is detected. Another gap is compliance drift: resources can become non-compliant after deployment because configurations, identities, permissions, or policies change over time.

The proposed research addresses these gaps by expressing compliance controls as code, validating infrastructure before deployment, integrating policy checks into CI/CD pipelines, and evaluating compliance across multiple cloud providers. The research is therefore positioned at the intersection of cloud security, governance, compliance automation, Infrastructure as Code, and DevSecOps.

IV. AIM AND OBJECTIVES

The aim of the research is to design and evaluate a compliance and governance by design framework that ensures proactive, consistent, and automated compliance enforcement across multi-cloud environments.

- Analyze regulatory and organizational compliance requirements relevant to multi-cloud adoption.
- Design a Policy as Code (PaC) framework that enforces compliance rules during resource provisioning.
- Integrate compliance checks into CI/CD pipelines using a DevSecOps approach.
- Implement and test the framework across AWS, Azure, and GCP.
- Evaluate the effectiveness of the framework in reducing compliance drift, improving audit readiness, and enhancing security posture.

V. PROPOSED RESEARCH METHODOLOGY

Although The proposed methodology follows an engineering and evaluation-oriented research design. It converts compliance requirements into machine-enforceable policies and integrates those policies with Infrastructure as Code and CI/CD workflows. The principal stages are requirement analysis, framework design, pipeline integration, multi-cloud deployment, and evaluation.

A. Requirement Analysis

Regulatory and organizational requirements such as GDPR, HIPAA, PCI-DSS, and ISO 27001 will be analyzed to identify security controls relevant to multi-cloud infrastructure. The controls will be organized into categories such as identity and access management, encryption, logging, monitoring, network security, and configuration management.

B. Framework Design

Policy as Code templates will be developed for selected security and compliance controls. Infrastructure as Code templates will be used to describe cloud resources. The framework will include validation mechanisms capable of evaluating configurations before deployment and identifying violations that require remediation.

C. CI/CD and DevSecOps Integration

Compliance validation will be integrated into CI/CD workflows using tools identified in the synopsis, including Open Policy Agent (OPA), Terraform Validator, Jenkins or GitLab CI/CD, and provider-specific policy services. A deployment should proceed only when required policy checks are satisfied, while violations should be reported for remediation.

D. Multi-Cloud Deployment

Validated Infrastructure as Code configurations will be used to provision representative workloads on AWS, Azure, and GCP. The experiments will consider equivalent security requirements across providers and examine how a common governance model can accommodate provider-specific implementation differences.

E. Evaluation

The framework will be evaluated using measurable indicators identified in the synopsis: Compliance Drift Detection Rate, Mean Time to Compliance (MTTC), Audit Readiness Score, and Deployment Latency Overhead. Where appropriate, baseline and framework-enabled deployment scenarios can be compared to determine the effect of automated compliance enforcement.

Table 2: Proposed Research Framework

Stage	Input	Activity	Expected Output
1. Requirement	GDPR/HIPAA/PCI-	Identify and classify	Compliance control

Analysis	DSS/ISO 27001 and organizational controls	required controls	catalogue
2. Framework Design	Security controls and IaC definitions	Develop PaC rules and validation structure	Policy library
3. Pipeline Integration	IaC repository and CI/CD platform	Integrate policy validation into pipeline	Automated compliance gate
4. Deployment	Validated templates	Provision representative workloads on AWS, Azure and GCP	Multi-cloud test environment
5. Evaluation	Compliance and deployment observations	Measure drift, MTTC, audit readiness and latency	Evaluation results
6. Refinement	Evaluation findings	Update policies and framework components	Reusable framework

VI. PROPOSED ARCHITECTURE

The proposed architecture consists of five logical layers. The first layer contains regulatory and organizational requirements. The second converts those requirements into Policy as Code and Infrastructure as Code. The third layer is the DevSecOps pipeline where policies are automatically validated. The fourth layer contains the target cloud environments: AWS, Azure, and GCP. The fifth layer provides continuous compliance monitoring, reporting, audit evidence, and remediation feedback.

The conceptual workflow is: Compliance Requirements → Policy as Code → Infrastructure as Code → CI/CD Validation → Multi-Cloud Provisioning → Continuous Compliance Monitoring → Audit and Remediation Feedback. This design makes compliance a part of the deployment lifecycle rather than a separate post-deployment activity.

VII. ALGORITHMIC RESEARCH PROCEDURE

Input: Multi-cloud infrastructure definitions, compliance requirements, and Policy as Code rules.

Output: Validated deployment configuration and compliance status.

Step 1: Collect applicable regulatory and organizational requirements.

Step 2: Map each requirement to one or more technical security controls.

Step 3: Express the controls as machine-readable Policy as Code rules.

Step 4: Define infrastructure using Infrastructure as Code templates.

Step 5: Execute policy validation during the CI/CD pipeline.

Step 6: Reject, flag, or route non-compliant configurations for remediation.

Step 7: Deploy validated resources to AWS, Azure, or GCP.

Step 8: Continuously monitor deployed resources for compliance drift.

Step 9: Record compliance evidence and evaluate MTTC, drift detection, audit readiness, and deployment overhead.

Step 10: Refine policies and controls based on evaluation findings.

VIII. EXPECTED RESULTS AND DISCUSSION

The supplied doctoral synopsis does not report completed experimental measurements. Therefore, this paper does not present fabricated numerical accuracy, percentages, or performance claims. The expected results are derived from the stated research objectives and methodology.

Table 3: Result Analysis

Research Objective	Expected Outcome	Evaluation Basis
Multi-cloud compliance analysis	Structured control catalogue mapped to regulatory requirements	Coverage and traceability of requirements
Policy as Code	Automated enforcement of selected security controls	Policy validation outcomes
CI/CD integration	Compliance checks before deployment	Pipeline pass/fail and remediation records

AWS/Azure/GCP implementation	Cross-provider demonstration of common governance principles	Successful validation across target platforms
Compliance drift reduction	Earlier identification of configuration violations	Compliance Drift Detection Rate
Audit readiness	Automated evidence and consistent control reporting	Audit Readiness Score
Operational efficiency	Controlled compliance overhead	MTTC and Deployment Latency Overhead

IX. RESEARCH SIGNIFICANCE

The proposed research is significant because it treats compliance as an engineering property of cloud infrastructure. Instead of considering governance only as an audit activity, the framework incorporates requirements into the infrastructure and software delivery lifecycle. This can support repeatable enforcement and reduce dependence on manual configuration reviews.

The research also addresses the practical problem of heterogeneous cloud environments. A multi-cloud organization may use different native security services and policy mechanisms on each provider. A common governance layer based on machine-readable policies can provide a consistent conceptual control model while allowing provider-specific implementation.

The proposed contribution is therefore expected to be relevant to cloud security, DevSecOps, Infrastructure as Code, policy automation, continuous compliance, and multi-cloud governance.

X. COMPARATIVE ANALYSIS OF SECURITY SETTINGS

Table 4: Comparative Analysis

Setting	Main Feature	Governance / Security Focus	Research Relevance
Traditional Infrastructure	Centrally managed infrastructure	Manual security configuration and audits	Baseline
Single Cloud	Provider-specific controls	Native policy and compliance services	Existing practice

Multi-Cloud	Heterogeneous platforms	Cross-provider governance and consistency	Primary problem setting
Policy as Code	Machine-readable rules	Automated policy validation and enforcement	Core mechanism
Compliance by Design	Controls integrated into lifecycle	Pre-deployment and continuous compliance	Proposed direction
DevSecOps Multi-Cloud	Security integrated into CI/CD	Automated compliance gates and feedback	Proposed implementation setting

XI. APPLICATION AND EVALUATION PERSPECTIVE

A practical application of the framework can use a representative enterprise workload containing compute, storage, databases, networking, identity roles, and logging services. Equivalent compliance requirements can be expressed at the control level and then mapped to the implementation mechanisms available on AWS, Azure, and GCP.

For example, an encryption requirement can be represented as a policy that requires encryption for selected storage resources. A least-privilege requirement can be represented through IAM rules. Logging requirements can require appropriate audit logging to be enabled. During deployment, the policy engine can evaluate the IaC configuration and identify violations before resources are provisioned.

The evaluation can use a baseline workflow without the proposed framework and a framework-enabled workflow. The comparison should focus on measurable observations rather than assumed improvements. Compliance drift can be measured by the proportion of introduced violations detected by the framework, while MTTC can capture the time from violation detection to compliant state.

XII. EXPECTED CONTRIBUTIONS

- A reusable compliance-by-design framework for heterogeneous multi-cloud environments.
- A Policy as Code library covering selected security and compliance controls.

- Integration of compliance validation into DevSecOps/CI/CD workflows.
- A multi-cloud implementation and evaluation across AWS, Azure, and GCP.
- An evaluation model based on compliance drift detection, MTTC, audit readiness, and deployment latency.
- A research contribution connecting cloud governance, Infrastructure as Code, Policy as Code, and continuous compliance.

XIII. CONCLUSION

This paper presented a structured research framework for a proactive approach to securing multi-cloud environments through Compliance and Governance by Design. The reviewed material indicates that cloud security involves shared responsibilities, heterogeneous controls, configuration management, identity and access management, monitoring, encryption, and regulatory compliance. The proposed research addresses the need to move compliance from predominantly reactive auditing toward automated, lifecycle-integrated enforcement.

Policy as Code, Infrastructure as Code, and DevSecOps provide the technical foundation for the proposed approach. The framework is intended to validate compliance before deployment, support consistent governance across AWS, Azure, and GCP, detect compliance drift, and generate evidence for audit readiness. Since the supplied synopsis does not contain completed experimental results, numerical performance claims are intentionally excluded. The final contribution will depend on implementation, rigorous evaluation, and analysis of the proposed framework.

REFERENCES

1. Kousalya A. and Baik Nam-kyun, "Enhance cloud security and effectiveness using improved RSA-based RBAC with XACML technique," *International Journal of Intelligent Networks*, vol. 4, pp. 62–67, Elsevier, 2023.
2. S. Batchu, "Cloud Infrastructure Fortification: Advanced Security Strategies in the Era of Emerging Threats," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 1, pp. 1407–1414, 2025.
3. M. Armbrust et al., "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
4. D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.
5. S. Harris et al., "CISSP: Certified Information Systems Security Professional Exam Guide," McGraw-Hill, 2020.
6. Cloud Security Alliance, "Cloud Controls Matrix," Cloud Security Alliance, 2017.
7. Y. Chen and K. Zhao, "Cloud Security Monitoring and Detection Approaches: A Survey," *Journal of Cloud Computing*, vol. 8, no. 1, 2019.

8. Y. Liu et al., "Compliance and Governance in Azure Cloud," *International Journal of Cloud Computing and Services Science*, vol. 7, no. 5, pp. 345–359, 2019.
9. Parker et al., "Open-source serverless architectures: an evaluation of Apache OpenWhisk," 2020.
10. F. Zaeem et al., "Network Security in Cloud: Leveraging Tools for Better Protection," *Proceedings of the International Conference on Cloud Computing and Big Data*, pp. 238–249, 2019.
11. Wenqing Qi, "Optimization of cloud computing task execution time and user QoS utility by improved particle swarm optimization," Elsevier, 2021.
12. R. Anderson et al., "Data Security and Cloud Computing: A Survey," Springer, 2016.
13. Liang et al., "A reliable trust computing mechanism based on multisource feedback and fog computing in social sensor cloud," 2020.
14. Shrestha et al., "Challenges of future VANET and cloud-based approaches," Wiley Online Library, 2018.
15. Goh et al., "Conceptual design of cloud-based data pipeline for smart factory," Springer, 2021.
16. Zhang et al., "Integration of on-premises and cloud-based software: the product bundling perspective," *Journal of the Association for Information Systems*, 2020.
17. Liu et al., "A cloud-edge collaboration framework for cognitive service," *IEEE Transactions on Cloud Computing*, 2020.
18. M. Jones et al., "Challenges in Cloud Security Configuration: A Review," *International Journal of Information Security*, vol. 19, no. 3, pp. 345–358, 2020.
19. Wang et al., "Distributed group coordination of multiagent systems in cloud computing systems using a model-free adaptive predictive control strategy," *IEEE Transactions on Cloud Computing*, 2021.
20. H. Tabrizchi and M. K. Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," *The Journal of Supercomputing*, vol. 76, pp. 9493–9532, 2020.