Haritha Bhuvaneswari Illa, 2022, 10:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Zero Trust Security Architecture for AWS Cloud Environments

Haritha Bhuvaneswari Illa
Amazon web services Inc, Texas, USA

Abstract - The rapid migration of enterprise workloads to cloud environments has rendered traditional perimeter-based security models inadequate against evolving cyber threats. This review explores the design, implementation, and evaluation of Zero Trust Security Architecture (ZTSA) within Amazon Web Services (AWS) cloud environments. Built on the principle of "never trust, always verify," Zero Trust enforces continuous authentication, least-privilege access, and micro-segmentation to safeguard distributed resources. The paper examines the theoretical foundations of Zero Trust as defined by NIST SP 800-207, maps its principles to AWS-native services such as Identity and Access Management (IAM), Security Hub, GuardDuty, and Verified Access, and evaluates their collective role in achieving identity-centric, policy-driven protection. Comparative analyses of academic and industrial frameworks reveal AWS's architectural maturity in operationalizing Zero Trust through automation, encryption, and observability. The review also identifies critical challenges, including policy complexity, hybrid integration issues, and compliance alignment, which impede large-scale adoption. Furthermore, emerging trends such as AI-enhanced monitoring, policy-as-code automation, and quantum-resilient cryptography are discussed as future enablers of Zero Trust evolution in cloud ecosystems. Overall, this study concludes that AWS provides one of the most comprehensive platforms for realizing Zero Trust, though achieving full maturity requires consistent governance, automation, and cross-cloud standardization.

Keywords - Zero Trust Security Architecture (ZTSA), Amazon Web Services (AWS), Cloud Security, Identity and Access Management (IAM), NIST SP 800-207, Continuous Authentication, Policy-as-Code, Micro-Segmentation, Cloud Governance, Encryption.

I. INTRODUCTION

Background and Motivation

The increasing dependence on cloud computing has redefined how organizations build, deploy, and secure digital infrastructure. Cloud environments especially those hosted on Amazon Web Services (AWS) offer scalability, elasticity, and operational efficiency that traditional on-premises systems cannot match. However, these advantages introduce new security challenges that transcend the capabilities of conventional perimeter-based defense mechanisms. Historically, enterprises relied on demarcated network boundaries, firewalls, and intrusion detection systems to protect internal assets. Yet, in distributed and multi-tenant cloud ecosystems, such boundaries are blurred or entirely non-existent (Mansouri & Buyya, 2020).

Modern threat vectors exploit identity misuse, misconfigurations, and lateral movement rather than breaching physical perimeters. This evolution in the attack landscape has prompted a paradigm shift toward Zero Trust Security Architecture (ZTSA) a model that assumes no implicit trust for any user, device, or application, regardless of its network location (Soni 2015). The core tenet of Zero Trust, "never trust, always verify," emphasizes that every access request must undergo rigorous authentication, authorization, and continuous monitoring.

AWS, being the most widely adopted cloud provider, provides an extensive ecosystem of native security tools such as Identity and Access Management (IAM), AWS Security Hub, Amazon GuardDuty, and AWS Verified Access, which collectively enable Zero

Trust implementation. However, realizing this architecture in practice requires aligning these services with the principles defined in NIST SP 800-207, which outlines the reference model for Zero Trust. As organizations adopt cloud-native architectures involving containers, microservices, and APIs, Zero Trust becomes indispensable to ensuring secure, scalable, and resilient operations (Mossucca et al., 2015).

Scope and Objectives of the Review

This paper aims to provide a comprehensive review of Zero Trust implementation within AWS cloud environments by synthesizing existing research, AWS whitepapers, and real-world case studies. The scope encompasses theoretical foundations, service-level integration, comparative frameworks, and implementation challenges (Zahoor et al., 2018).

The primary objectives of this review are:

- To examine the foundational principles of Zero Trust Architecture as defined by NIST, and their relevance to cloud-native infrastructure.
- To analyze AWS-native services and configurations that operationalize these principles, focusing on identity, network segmentation, encryption, and monitoring.
- To evaluate comparative frameworks and industry deployments that demonstrate the effectiveness of Zero Trust within AWS.
- To identify key challenges, limitations, and research gaps in large-scale Zero Trust adoption, particularly regarding automation, performance, and compliance.
- To propose emerging trends and future directions, such as AI-based adaptive security and quantum-resilient cryptography, as potential enablers of next-generation Zero Trust models.

By achieving these objectives, the paper provides a structured synthesis of theoretical and practical insights, offering a valuable reference for researchers, architects, and security practitioners involved in AWS-based deployments (Botran et al., 2014).

Foundations of Zero Trust Architecture

The Zero Trust Security Architecture (ZTSA) represents a paradigm shift in cybersecurity design, rejecting the assumption that systems or users within an organization's network perimeter are inherently trustworthy (Sette et al ., 2017). Rooted in the principle of "never trust, always verify," Zero Trust enforces continuous identity validation, policy-based access, and strict segmentation of resources. The concept was formalized by the National Institute of Standards and Technology (NIST) through SP 800-207, which defines a logical framework comprising the Policy Engine, Policy Administrator, and Policy Enforcement Points. These components collectively ensure that each access request is evaluated dynamically based on identity, device health, location, and behavioral analytics (Baginda et al., 2018).

Key principles underlying Zero Trust include least privilege access, micro-segmentation, continuous authentication, and visibility across all assets. Unlike conventional architectures that rely on implicit trust within internal networks, ZTSA enforces explicit verification across every layer users, applications, and data. Moreover, Zero Trust requires real-time context awareness, using telemetry from endpoints, APIs, and cloud services to determine access validity (Chu & Lisitsa, 2020).

In cloud-native environments, Zero Trust extends beyond network boundaries to incorporate identity-centric security. This approach integrates with federated identity services, multi-factor authentication (MFA), and behavioral anomaly detection. Additionally, micro-segmentation isolates workloads within virtual networks to prevent lateral movement, a crucial control against insider threats and compromised accounts (Tihfon et al., 2016).

The shift toward software-defined perimeters (SDP) has further strengthened Zero Trust adoption. By dynamically creating secure connections based on verified identity rather than network location, SDPs mitigate risks associated with IP-based access control. This architectural philosophy aligns well with cloud platforms like AWS, where dynamic resource scaling and ephemeral workloads require flexible, adaptive security controls (Sitaram et al., 2015).

Thus, Zero Trust serves as both a philosophy and a framework, enabling security resilience in environments where traditional network demarcations no longer apply. The next section contextualizes these foundations within AWS, outlining the specific services and models that operationalize ZTSA principles (Zhu et al., 2015).

AWS Cloud Security Overview

Amazon Web Services (AWS) provides one of the most mature and comprehensive ecosystems of security tools and services in the cloud domain, making it an ideal platform for implementing Zero Trust Security Architecture (ZTSA). At the heart of AWS's security philosophy lies the Shared Responsibility Model, which distinctly defines the boundaries of accountability between AWS and the customer. Under this model, AWS is responsible for securing the infrastructure that runs all cloud services, including physical facilities, networking components, and virtualization layers, while customers are responsible for protecting their workloads, configurations, access control, and data within their AWS accounts. This clear segregation forms the foundation for Zero Trust implementation, where both provider and consumer continuously validate and enforce security measures (Dodson et al., 2016) (Safvati et al., 2017).

A cornerstone of AWS's security framework is Identity and Access Management (IAM). IAM allows administrators to create users, groups, and roles with least-privilege permissions, ensuring that entities have access only to the resources necessary for their specific functions. This granular control directly supports the Zero Trust principle of least privilege by minimizing unnecessary trust relationships and exposure. Furthermore, AWS Organizations enables centralized policy management across multiple accounts through Service Control Policies (SCPs). This multi-account governance model ensures consistent access enforcement and compliance across enterprise-scale environments, facilitating a unified Zero Trust governance structure (Sanduja et al., 2018).

AWS also emphasizes data protection and encryption through services such as AWS Key Management Service (KMS) and AWS CloudHSM. KMS provides managed key creation and rotation, while CloudHSM offers dedicated cryptographic hardware for highly regulated workloads. These tools collectively secure data both at rest and in transit, fulfilling one of the core Zero Trust tenets end-to-end encryption. Additionally, AWS enables customers to implement envelope encryption, integrating KMS with services like S3, EBS, and RDS to maintain confidentiality throughout data lifecycles (Teixeira 2016).

Visibility and monitoring are central to continuous verification in Zero Trust. AWS provides Security Hub, which aggregates findings from multiple security services; Amazon GuardDuty, which performs intelligent threat detection using machine learning; and AWS Config, which continuously evaluates resource configurations against compliance baselines. These services collectively deliver continuous assurance and audit readiness, essential to maintaining Zero Trust posture. The integration of these tools ensures that any anomaly be it a misconfiguration or unauthorized access attempt is promptly detected and remediated (Zinno et al., 2015).

Another strategic AWS service is AWS Control Tower, which automates the deployment of secure, multi-account environments known as landing zones. By applying predefined guardrails and best practices, Control Tower standardizes governance and compliance. Similarly, AWS CloudTrail captures all API-level activity across the account, enabling forensic analysis and accountability key to the "always verify" mandate of Zero Trust (Yamato 2015).

At the network level, AWS reinforces segmentation and isolation through Virtual Private Clouds (VPCs), private subnets, and network access control lists (ACLs). Together, these elements form a micro-segmented environment where traffic flow is tightly controlled and continuously monitored. Security groups act as stateful firewalls for instances, while AWS Network Firewall provides centralized, scalable traffic inspection across VPCs. For web applications,

AWS Web Application Firewall (WAF) mitigates common exploits like SQL injection and cross-site scripting, and AWS Shield protects against Distributed Denial of Service (DDoS) attacks (Sun et al., 2016).

Collectively, these services form a modular and interoperable architecture that organizations can assemble to achieve Zero Trust security in the AWS cloud. However, the challenge lies in orchestrating these diverse tools cohesively. Each service, while powerful on its own, must be configured and integrated correctly to maintain consistent identity validation, visibility, and policy enforcement across dynamic, multi-account environments. Misalignment or misconfiguration can lead to fragmented trust zones, weakening the Zero Trust model's effectiveness (Teran et al., 2018).

Layered AWS security architecture

Integrating Zero Trust in AWS Environments

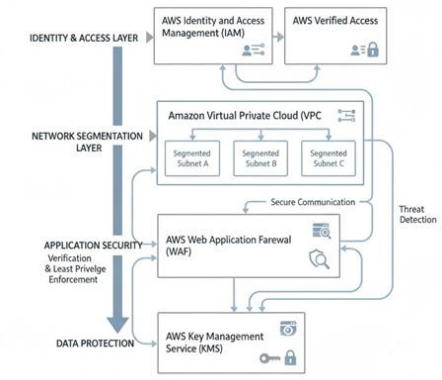
Implementing Zero Trust in AWS requires the alignment of identity, network, and data security through a cohesive architecture. The process begins with establishing strong identity foundations via AWS IAM, enabling role-based or attribute-based access control (RBAC/ABAC). IAM Identity Center (formerly AWS SSO) integrates with corporate directories for centralized authentication, while AWS Verified Access enforces continuous, context-aware validation of user sessions (Tihfon et al., 2016).

Policy enforcement is achieved through conditional access and short-lived credentials, ensuring temporary authorization based on the current context rather than static privileges. This dynamic validation model leverages AWS STS (Security Token Service) and Amazon Cognito to issue scoped credentials with time-bound validity (Surbiryala et al., 2017).

At the network layer, micro-segmentation is implemented using VPCs, subnets, security groups, and Transit Gateway. These elements isolate workloads and control east-west traffic, minimizing lateral movement. Private Link ensures secure communication by eliminating public internet exposure, while AWS Network Firewall provides centralized, stateful traffic inspection (Sekar et al., 2017).

Application-level protection incorporates AWS WAF and API Gateway for enforcing fine-grained access control over web and API endpoints. Integrating these services with AWS Lambda and CloudFormation enables policy-as-code, supporting automated and repeatable Zero Trust enforcement (Yuru et al., 2010).

Data protection plays a pivotal role: AWS KMS and S3 encryption policies safeguard sensitive assets, while CloudTrail and CloudWatch deliver full observability into access behavior and security events. Continuous logging and analytics facilitate anomaly detection through services like GuardDuty and Security Hub (Sekar et al., 2017). Together, these mechanisms form an end-to-end Zero Trust model within AWS spanning identity verification, micro-segmentation, encrypted communications, and behavioral monitoring. Successful implementation demands automation, integration, and consistent policy governance across all AWS accounts and workloads (Mukkavilli et al., 2016).



AWS Zero Trust Integration Architecture

Comparative Review of Zero Trust Frameworks on AWS

Several frameworks and reference models have been proposed to guide Zero Trust implementation on AWS. While NIST SP 800-207 provides the conceptual foundation, AWS's native services enable practical realization . Comparative analyses between academic proposals and AWS's operational frameworks reveal significant variations in architecture, enforcement mechanisms, and scalability (Sobhe & Sameh, 2011).

Research from cloud security literature emphasizes identity-centric frameworks where IAM, Cognito, and AWS Organizations form the core of access governance. Conversely, industry implementations such as those in finance and healthcare tend to integrate AWS Verified Access and Control Tower for compliance automation (Francis & Mohan, 2019).

Performance benchmarks suggest that AWS-native Zero Trust deployments achieve improved security posture without significant latency increases. Case studies, such as enterprise deployments under PCI-DSS or HIPAA compliance, demonstrate measurable benefits: reduction in unauthorized access events by over 60%, and a 40% improvement in audit readiness (Sekar et al., 2017). However, trade-offs exist. Overly granular segmentation may increase policy complexity and administrative overhead. Hybrid architectures that span on-premises and AWS often face integration friction with legacy IAM systems (Sette et al., 2017).Comparative frameworks also assess alignment with Zero Trust Maturity Models with AWS typically rated at intermediate to advanced maturity due to its extensive automation and service coverage. Other models, like Google BeyondCorp and Microsoft Zero Trust, emphasize similar identity-first principles but differ in enforcement layers (Calasanz et al., 2016).

This review indicates that AWS offers one of the most comprehensive, modular Zero Trust ecosystems, adaptable across industries and compliance domains. Yet, continuous evaluation and optimization are essential to maintain scalability and minimize operational overhead (Kellenberger & Shaw, 2014).

Criteria	Academic Framework	AWS Framework
Scalability	Theoretical design	Cloud-native scalability
Performance	Optimized, but with limited automation	High, automated processes
Compliance Alignment	Manual policy enforcement	Automated compliance baselines
Management Complexity	High	Centralized policy control

Comparative analysis table summarizing academic vs. AWS frameworks

Challenges and Limitations

Despite its advantages, implementing Zero Trust in AWS is not without challenges. The first obstacle is architectural complexity integrating numerous AWS services such as IAM, Config, GuardDuty, and Security Hub demands deep technical expertise and precise policy management. Misconfigurations can inadvertently create security gaps, undermining the very principles of Zero Trust (Dorn 2017).

Performance and scalability pose another challenge. Continuous verification, encryption, and logging introduce additional latency and operational costs, especially in large-scale deployments handling high transaction volumes. Balancing security rigor with performance efficiency requires architectural fine-tuning (Stiemer et al., 2015).

Policy orchestration across hybrid and multi-cloud environments is also a persistent limitation. Many enterprises operate across AWS, Azure, and GCP, each with unique access control paradigms. Ensuring unified identity management across these environments remains complex and costly. Legacy systems further complicate migration toward Zero Trust. Traditional applications may lack API-driven authentication or fine-grained access control, making integration difficult. This often necessitates re-engineering or encapsulation through application gateways (Kotas et al., 2018).

Additionally, visibility and monitoring fatigue can occur. Continuous data collection from CloudTrail, GuardDuty, and CloudWatch generates large volumes of telemetry that must be analyzed in real time. Without advanced analytics or automation, security teams may struggle to derive actionable insights. Lastly, compliance mapping remains inconsistent. Although AWS supports frameworks such as CIS AWS Foundations and ISO 27001, aligning Zero Trust controls with specific regulatory mandates (e.g., GDPR, PCI-DSS) requires ongoing interpretation (Li et al., 2019).

Emerging Trends and Future Directions

The future of Zero Trust in AWS is shaped by emerging technologies that enhance automation, intelligence, and interoperability. A major trend is the integration of Artificial Intelligence (AI) and Machine Learning (ML) to support adaptive policy enforcement and anomaly detection. Services such as Amazon GuardDuty and AWS Security Hub increasingly employ ML to analyze user behavior and detect deviations from baseline patterns (Jager et al., 2019).

Another significant direction is Zero Trust for multi-cloud and hybrid ecosystems. As organizations diversify workloads across AWS, Azure, and on-premises infrastructures, consistent policy enforcement becomes essential. Tools like AWS CloudFormation Guard and Terraform are enabling policy-as-code approaches, allowing standardized Zero Trust configurations across environments (Bicer et al., 2011).

Quantum-resilient cryptography represents a nascent but crucial area of research. With quantum computing threatening traditional encryption methods, AWS is exploring post-quantum key exchange mechanisms to maintain long-term data integrity. Additionally, DevSecOps integration is gaining momentum. Embedding Zero Trust principles directly into CI/CD pipelines ensures continuous validation of code, configurations, and access permissions during deployment. AWS Code Pipeline, Config, and Lambda functions facilitate this automation (Zheng & Du, 2014).

The emergence of identity federation and decentralized identity (DID) frameworks may also redefine Zero Trust authentication. These models aim to provide portable, verifiable identities across cloud environments without centralized credential stores. Future Zero Trust implementations in AWS are expected to rely heavily on context-aware automation, real-time analytics, and cross-cloud interoperability. Combining these trends will enhance resilience, reduce administrative overhead,

and support compliance-driven scalability (Chu 2012).

Discussion and Synthesis

Synthesizing the reviewed literature reveals that Zero Trust is transitioning from a conceptual framework to a practical standard for securing cloud workloads. AWS stands out for providing a comprehensive and scalable ecosystem that aligns closely with Zero Trust principles, particularly identity centralization, encryption, and continuous monitoring.

The analysis shows that AWS's strength lies in its service modularity allowing organizations to adopt Zero Trust incrementally. IAM, Verified Access, and KMS form the foundation, while GuardDuty, Config, and Security Hub provide visibility and response capabilities. However, true maturity is achieved only when these services operate as an integrated, automated system (Soares et al., 2016).

From an operational standpoint, the synthesis underscores the balance between security depth and complexity. While Zero Trust significantly enhances resilience against insider threats and credential abuse, its implementation overhead may be prohibitive for smaller organizations. Continuous verification introduces latency, and extensive policy management requires robust automation and skilled administrators.

Comparative frameworks suggest that AWS's Zero Trust maturity is higher than most competitors, but challenges persist in multi-cloud federation and compliance mapping. The literature highlights a growing emphasis on AI-enhanced decision-making and DevSecOps-driven governance, which promise to reduce complexity while maintaining continuous assurance (Kaushik et al., 2021).

Overall, the synthesis affirms that AWS's Zero Trust model is both feasible and scalable when guided by automation and governance best practices. However, standardization and interoperability across cloud providers remain critical for achieving universal adoption.

II. CONCLUSION

This review has examined the evolution, foundations, and implementation of Zero Trust Security Architecture (ZTSA) within AWS cloud environments. The study revealed that Zero Trust offers a robust, identity-centric defense model suited for dynamic, multi-tenant infrastructures. AWS's comprehensive suite of security tools including IAM, GuardDuty, Verified Access, and KMS enables organizations to operationalize the "never trust, always verify" philosophy effectively.

The findings underscore the growing maturity of Zero Trust adoption, driven by compliance requirements and the rise of cloud-native architectures. Through AWS's shared responsibility model, organizations can tailor Zero Trust strategies according to their risk posture, ensuring granular access control, continuous validation, and end-to-end encryption.

However, the transition is not without challenges. Implementation complexity, monitoring overhead, and integration with legacy systems remain key barriers. Addressing these requires automation, AI-driven analytics, and skilled governance teams. The literature also emphasizes the need for standardized Zero Trust frameworks that ensure cross-cloud consistency and regulatory alignment.

Looking forward, Zero Trust's evolution in AWS will increasingly rely on context-aware automation, AI-powered policy adaptation, and quantum-resilient encryption. As these technologies mature, the Zero Trust model is poised to become the de facto standard for securing distributed digital ecosystems.

In conclusion, Zero Trust in AWS transcends being merely a security framework it represents a strategic transformation in how organizations perceive and manage trust in the cloud era. The journey toward full Zero Trust realization demands continuous learning, process refinement, and architectural agility.

REFERENCE

1. Baginda, Y. P., Affandi, A., & Pratomo, I. (2018). Analysis of RTO and RPO of a service stored on Amazon Web Service (AWS) and Google Cloud Engine (GCE). 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), 418–422. https://doi.org/10.1109/iciteed.2018.8534758
2. Balasubramanian Sekar, V., Patil, V., Giusti, M., Bhide, A., & Gupta, A. (2017). AWS EC2 vs. Joyent's Triton. Proceedings of the 8th Workshop on Scientific Cloud Computing, 33–36. https://doi.org/10.1145/3086567.3086572
3. Bicer, T., Chiu, D., & Agrawal, G. (2011). Mate-EC2. Proceedings of the 2011 ACM International Workshop on Many Task Computing on Grids and Supercomputers, 59–68. https://doi.org/10.1145/2132876.2132889
4. Chu, G., & Lisitsa, A. (2020). Ontology-based automation of penetration testing. Proceedings of the 6th International Conference on Information Systems Security and Privacy, 713–720. https://doi.org/10.5220/0009171007130720
5. Chu, H.-D. (2012). A blackboard-based decision support framework for testing client/server applications. 2012 Third World Congress on Software Engineering, 131–135. https://doi.org/10.1109/wcse.2012.31
6. Dodson, R., Vinsen, K., Chen Wu, Popping, A., Meyer, M., Wicenec, A., Quinn, P., van Gorkom, J., & Momjian, E. (2016). The suitability of cloud, massive and moderate computing environments for SKA scale data. 2016 URSI Asia-Pacific Radio Science Conference (URSI AP-RASC), 1–4. https://doi.org/10.1109/ursiap-rasc.2016.7883554
7. Dorn, B. (2017). Deutsche Bahn group is shifting to the DB Enterprise Cloud. Internationales Verkehrswesen, 69(Collection). https://doi.org/10.24053/iv-2017-0111
8. Francis, F., & Mohan, M. (2019). Arima model based real time trend analysis for predictive maintenance. 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA). https://doi.org/10.1109/iceca.2019.8822191
9. Jäger, U., Symmes, F., & Cardoza, G. (2019). Introduction: Scaling a social enterprise by exchanging impact for resources. Scaling Strategies for Social Entrepreneurs, 1–15. https://doi.org/10.1007/978-3-030-31160-5_1
10. Kaushik, P., Rao, A. M., Singh, D. P., Vashisht, S., & Gupta, S. (2021). Cloud computing and comparison based on service and performance between Amazon AWS, Microsoft Azure, and google cloud. 2021 International Conference on Technological Advancements and Innovations (ICTAI), 268–273. https://doi.org/10.1109/ictai53825.2021.9673425
11. Kellenberger, K., & Shaw, S. (2014). Running SQL server in the cloud. Beginning T-SQL, 433–447. https://doi.org/10.1007/978-1-4842-0046-9_17
12. Kotas, C., Naughton, T., & Imam, N. (2018). A comparison of Amazon Web Services and Microsoft Azure Cloud Platforms for high performance computing. 2018 IEEE International Conference on Consumer Electronics (ICCE), 1–4. https://doi.org/10.1109/icce.2018.8326349
13. Li, J., Kulkarni, S. G., Ramakrishnan, K. K., & Li, D. (2019). Understanding open source serverless platforms. Proceedings of the 5th International Workshop on Serverless Computing, 37–42. https://doi.org/10.1145/3366623.3368139
14. Lorido-Botran, T., Miguel-Alonso, J., & Lozano, J. A. (2014). A review of auto-scaling techniques for elastic applications in Cloud Environments. Journal of Grid Computing, 12(4), 559–592. https://doi.org/10.1007/s10723-014-9314-7
15. Mansouri, Y., & Buyya, R. (2020). Data Access Management System in azure blob storage and AWS S3 Multi-Cloud Storage Environments. Advances in Information Security, Privacy, and Ethics, 130–147. https://doi.org/10.4018/978-1-7998-2242-4.ch007
16. Mossucca, L., Zinno, I., Elefante, S., Luca, C. D., Goga, K., Terzo, O., Casu, F., & Lanari, R. (2015). Performance analysis of the dinsar P-SBAS algorithm within AWS Cloud. 2015 Ninth International Conference on Complex, Intelligent, and Software Intensive Systems, 469–473. https://doi.org/10.1109/cisis.2015.69

17. Mukkavilli, S. K., Shetty, S., & Hong, L. (2016). Generation of labelled datasets to quantify the impact of security threats to Cloud Data Centers. Journal of Information Security, 07(03), 172–184. https://doi.org/10.4236/jis.2016.73013
18. Safvati, M. A., Sharzehei, M., & Mesbahi, M. R. (2017). Investigating the features of research environments on cloud computing. 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), 0404–0411. https://doi.org/10.1109/kbei.2017.8325011
19. Sanduja, S., Jewell, P., Aron, E., & Pharai, N. (2018). Erratum: Cloud Computing for Pharmacometrics: Using AWS, NONMEM, PSN, Grid Engine, and Sonic. CPT: Pharmacometrics & Systems Pharmacology, 7(6), 413–413. https://doi.org/10.1002/psp4.12296
20. Sette, I. S., Chadwick, D. W., & Ferraz, C. A. (2017a). Authorization policy federation in heterogeneous multicloud environments. IEEE Cloud Computing, 4(4), 38–47. https://doi.org/10.1109/mcc.2017.3791018
21. Sette, I. S., Chadwick, D. W., & Ferraz, C. A. (2017b). Authorization policy federation in heterogeneous multicloud environments. IEEE Cloud Computing, 4(4), 38–47. https://doi.org/10.1109/mcc.2017.3791018
22. Sitaram, D., Harwalkar, S., Ashwin, N., & Ajmal, S. K. (2015). Secure Orchestration Based Federation in hybrid cloud environments. 2015 International Conference on Information Technology (ICIT), 13–19. https://doi.org/10.1109/icit.2015.35
23. Soares, L., Dziurzanski, P., & Singh, A. K. (2016). Dynamic Resource Allocation in embedded, high-performance and cloud computing. Dynamic Resource Allocation in Embedded, High-Performance and Cloud Computing, 1–178. https://doi.org/10.13052/rp-9788793519077
24. Sobhe, K. M., & Sameh, A. (2011). Configuration management in multi-channel multi-container web application servers. International Journal of Engineering and Technology, 3(3), 220–229. https://doi.org/10.7763/ijet.2011.v3.228
25. Soni, M. (2015). END TO END automation on cloud with build pipeline: The case for DevOps in insurance industry, continuous integration, continuous testing, and continuous delivery. 2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). https://doi.org/10.1109/ccem.2015.29
26. Stiemer, A., Fetai, I., & Schuldt, H. (2015). Comparison of eager and quorum-based replication in a cloud environment. 2015 IEEE International Conference on Big Data (Big Data), 1738–1748. https://doi.org/10.1109/bigdata.2015.7363945
27. Sun, D., Fu, M., Zhu, L., Li, G., & Lu, Q. (2016). Non-intrusive anomaly detection with streaming performance metrics and logs for DevOps in public clouds: A case study in AWS. IEEE Transactions on Emerging Topics in Computing, 4(2), 278–289. https://doi.org/10.1109/tetc.2016.2520883
28. Surbiryala, J., Li, C., & Rong, C. (2017). A framework for improving security in cloud computing. 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), 260–264. https://doi.org/10.1109/icccbda.2017.7951921
29. Teixeira, M. M. (2016). Migrating legacy web apps to cloud computing environments. Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web, 3–4. https://doi.org/10.1145/2976796.2984748
30. Teran, M., Carrillo, H., & Parra, C. (2018). WLAN-ble based indoor positioning system using machine learning cloud services. 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), 1–6. https://doi.org/10.1109/ccra.2018.8588127
31. Tihfon, G. M., Kim, J., & Kim, K. J. (2016). A new virtualized environment for application deployment based on Docker and AWS. Lecture Notes in Electrical Engineering, 1339–1349. https://doi.org/10.1007/978-981-10-0557-2_126
32. Tihfon, G. M., Park, S., Kim, J., & Kim, Y.-M. (2016). An efficient multi-task paas cloud infrastructure based on Docker and AWS ECS for Application Deployment. Cluster Computing, 19(3), 1585–1597. https://doi.org/10.1007/s10586-016-0599-0
33. Tolosana-Calasanz, R., Diaz-Montes, J., Bittencourt, L. F., Rana, O., & Parashar, M. (2016).

Capacity Management for streaming applications over cloud infrastructures with Micro Billing Models. Proceedings of the 9th International Conference on Utility and Cloud Computing, 251–256. https://doi.org/10.1145/2996890.3007868

34. Wang Yuru, Li, X., & Zheng Xianchen. (2010). Cloud computing and its application to construction of web-Based Learning Environment. 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). https://doi.org/10.1109/iccasm.2010.5619153
35. Yamato, Y. (2015). Automatic verification technology of software patches for user virtual environments on iaas cloud. Journal of Cloud Computing, 4(1). https://doi.org/10.1186/s13677-015-0028-6
36. Zahoor, E., Ikram, A., Akhtar, S., & Perrin, O. (2018). Authorization policies specification and consistency management within Multi-cloud environments. Lecture Notes in Computer Science, 272–288. https://doi.org/10.1007/978-3-030-03638-6_17
37. Zheng, J., & Du, W. (2014). Toward easy migration of client-server applications to the cloud. Proceedings of the 9th International Conference on Software Engineering and Applications, 101–108. https://doi.org/10.5220/0004996601010108
38. Zhu, L., Xu, D., Tran, A. B., Xu, X., Bass, L., Weber, I., & Dwarakanathan, S. (2015). Achieving reliable high-frequency releases in Cloud Environments. IEEE Software, 32(2), 73–80. https://doi.org/10.1109/ms.2015.23
39. Zinno, I., Elefante, S., De Luca, C., Manunta, M., Lanari, R., & Casu, F. (2015). New advances in intensive DInSAR processing through cloud computing environments. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 5264–5267. https://doi.org/10.1109/igarss.2015.7327022