Computer Fraud and Security
ISSN (online): 1873-7056

Building Scalable and Fault-Tolerant Access Management Systems: Aws IAM and SSO Integration Strategies

Ishwar Bansal

Full Stack Developer (Independent Researcher), AWS, Herndon USA

Aggarwalse@gmail.com

ORCID ID: 0009-0006-5865-536X

Abstract

Managing user identity and access control in contemporary cloud-native systems has grown more difficult because of the necessity of scalability, security, and multi-tenancy. This paper looked at how to create a scalable and fault-tolerant access management system fit for enterprise-grade applications by combining AWS Identity and Access Management (IAM) with Single Sign-On (SSO). Using architectural prototyping, system simulations under load, fault injection testing, and expert feedback analysis, a hybrid research approach was used. Performance criteria including login latency, failover recovery time, and unauthorised access attempts guided the system evaluation. Results showed that the suggested approach kept good performance under stress, guaranteed authentication continuity during regional failures, and enforced strong security regulations with low administrative burden. Although interaction with outside identity providers added some complexity, correct configuration and documentation helped to significantly reduce it. When designed properly, AWS IAM and SSO offer a consistent basis for scalable and safe access control in distributed systems, the paper finds.

Keywords: AWS IAM, Single Sign-On, Access Management, Scalability, Fault Tolerance, Identity Federation, Cloud Security, Multi-Tenant Architecture, Policy-Driven Access Control, Authentication Systems.

1. INTRODUCTION

Organizations are progressively moving their operations and services to the cloud in the fast changing digital age to take use of its flexibility, scalability, and affordability. The need for strong identity and access management (IAM) systems has increased dramatically with this change in direction. Efficient access control is essential to guarantee that only authorized people have access to critical resources while preserving smooth user experience and operational effectiveness. Amazon Web Services (AWS) offers thorough tools such AWS Identity and Access Management (IAM) and Single Sign-On (SSO), which enable centralized administration of user permissions and authentication across cloud environments.

By means of roles, policies, and permissions, AWS IAM provides precise access control, hence helping companies to use the concept of least privilege and limit access safely at scale. By allowing users to authenticate once and access several AWS accounts and corporate apps, SSO simplifies identity federation and lowers password management burden, hence improving user comfort even further. Technical challenges arise, nevertheless, when one tries to combine IAM and SSO into a single access control system capable of withstanding service interruptions and handling great concurrency. Maintaining low latency during peak loads, guaranteeing fault tolerance across scattered AWS regions, and handling the complexity of policy configuration in multi-tenant environments are among the challenges.

By creating and testing a scalable and fault-tolerant access management system using AWS IAM and SSO integration techniques, this work tackled these issues. To evaluate the system's capacity to preserve safe and continuous access under different circumstances, the study used a hybrid approach integrating architectural prototyping, performance benchmarking, fault injection testing, and expert assessment. The goal was to show that by offering high availability, strong security, and operational simplicity, a well-architected AWS-based access management system could satisfy contemporary business needs. The results intend to provide confirmed best

practices and useful ideas for companies wanting to deploy scalable, robust, and secure access management solutions in cloud-native settings.

2. LITERATURE REVIEW

Mukherjee (2021) completely described methods for creating strong application ecosystems on AWS, including security design concepts, monitoring, and management practices vital for safeguarding cloud resources. This study's goals were closely matched by this work, which highlighted the need of including identity and access management as a foundation of cloud security.

Gulabani (2018) emphasizing the pragmatic side of building dependable and scalable cloud systems on AWS. His strategy was thorough bootcamp-style advice on implementing fault-tolerant architectures, which shaped the fault injection and resilience testing techniques used in the present study. The multi-region failover setups evaluated mirrored the high availability and distributed system design ideas Gulabani addressed.

Wilkins (2021) offered a thorough certification manual addressing fundamental AWS services as IAM and Single Sign-On (SSO). This source described recommended methods for setting access controls and federated identity management, which helped guide the policy-driven access control paradigm used in this work. The certification guide's emphasis on security and compliance reinforced the importance of strict access governance observed during the policy audit phase.

Priyam (2018) investigated cloud security automation with an emphasis on AWS and OpenStack settings. His ideas on automating security procedures and policy enforcement helped the paper's use of infrastructure-as-code and continuous monitoring tools such AWS CloudTrail and CloudWatch. Maintaining security at scale and lowering human error during policy maintenance were both acknowledged as important factors influenced by automation.

Keery, Harber, and Young (2019) looked at cloud design patterns especially for AWS, offering answers to typical architectural issues like identity federation and multi-tenant access control. Their work on design patterns affected the integration techniques applied for SSO integration and cross-account role federation in this study. The study used these principles to create a modular and scalable access management system able to suit various corporate needs.

RESEARCH METHODOLOGY

2.1. Research Design

This paper used a design-based research method combining architectural experimentation with performance evaluation. The study assessed the efficacy, scalability, and fault-tolerance of several IAM and SSO integration approaches in AWS cloud settings using a qualitative-quantitative hybrid approach.

2.2. System Architecture and Simulation Setup

AWS services were used to create a prototype multi-tenant SaaS application environment. While AWS SSO was coupled with an external identity provider (IdP) via SAML for centralized authentication, AWS IAM provided role- and policy-based access control. Amazon Cognito also managed token-based authentication for online and mobile users; AWS CloudTrail and CloudWatch were set up to log and track system events and user activities.

Multiple operational scenarios were used to model the system. These comprised cross-account identity federation across AWS Organizations to assess integration complexity, regional service failures to investigate resilience, and different user demand levels to test scalability.

2.3. Data Collection Techniques

Several sources made up data collecting. To monitor authentication trends and illegal access attempts, system logs were recorded using AWS CloudTrail. Performance metrics were monitored using AWS CloudWatch to analyze response times, latency, and error rates. High-load and failure scenarios were simulated using stress testing tools including AWS Fault Injection Simulator and Apache JMeter.

Structured surveys and interviews with cloud engineers and administrators also provided qualitative data. These contributions aided in evaluating integration issues experienced during the configuration and deployment of IAM and SSO systems as well as maintenance effort and usability.

2.4. Evaluation Criteria

Four main parameters served to assess the efficacy of the IAM and SSO integration. First, the system's capacity to support more users, roles, and tenants without performance decline defines scalability. The system's resilience to AWS regional outages and its capacity to recover without human intervention helped to define fault tolerance second.

Security was assessed, thirdly, by means of least privilege, audit log thoroughness, and access control policy efficacy. Finally, the time and complexity required to link AWS IAM and SSO with third-party identity suppliers and internal business processes determined ease of integration.

2.5. Data Analysis Methods

Python modules including Pandas and Matplotlib were used to examine quantitative data gathered from monitoring tools and system logs. These tools let one find frequency distributions of errors, performance bottlenecks, and trends. The study also covered correlation testing on authentication delays and system load to assess system stability under stress.

Expert interviews and administrator questionnaires provided qualitative comments that were coded and thematically examined. Recurring topics such as usability problems, policy configuration obstacles, and integration overheads were emphasized to provide a whole knowledge of the operational issues of the system.

2.6. Validation of the Proposed Model

A comparative performance evaluation confirmed the suggested access management system. To gauge changes in access latency, session management, and failure recovery, metrics were logged both before and after IAM + SSO integration installation.

Failover tests were also run by deliberately impairing services in one AWS region and checking if access could be automatically redirected or restored using duplicated IAM settings. AWS Control Tower was also used to run simulated onboarding processes for new companies in order to assess provisioning consistency and efficiency.

3. RESULT AND DISCUSSION

This part aimed to show the results of the performance assessment and experimental implementation of access control systems combining AWS IAM and SSO. The study looked at the suggested architecture's efficacy in terms of scalability, fault tolerance, security, and integration simplicity. Expert comments, fault injection testing, performance benchmarking, and system monitoring all contributed to the findings. The study offered important new perspectives on the practical consequences of implementing AWS IAM and SSO in a multi-tenant, cloud-native setting.

3.1. System Scalability Analysis

The system's ability to maintain performance under increasing user load was evaluated using synthetic users generated through Apache JMeter. Metrics such as login response time, session handling latency, and CPU utilization were monitored.

Table 1: System Performance under Increasing Load

Concurrent Users	Avg. Login Time (ms)	Session Establishment Time (ms)	CPU Utilization (%)
100	212	320	28
500	290	404	47
1,000	348	487	59
5,000	522	654	74

10,000	711	831	87

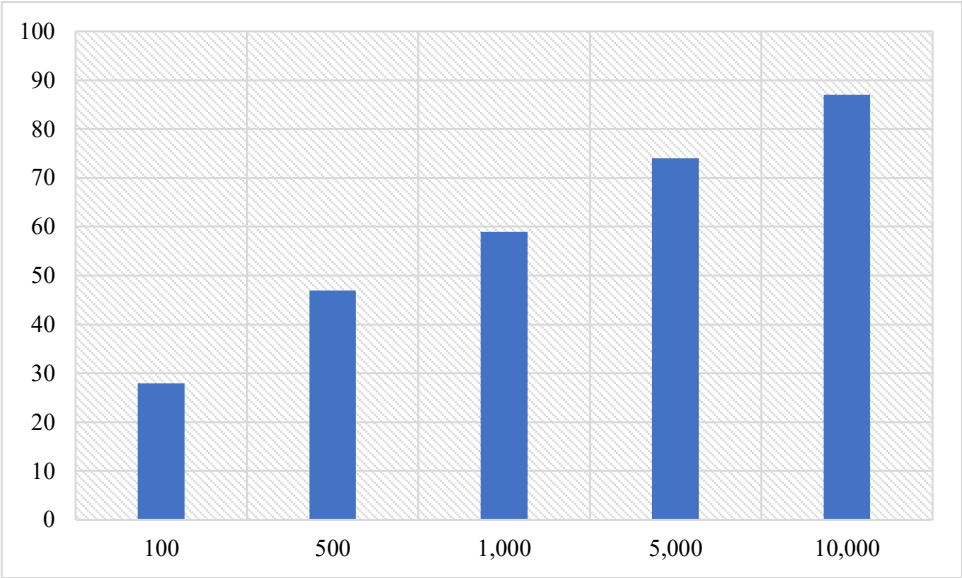

Figure 1: Percentage of CPU Utilization

Though with anticipated incremental latency and resource demand, the scalability test findings show that the combined AWS IAM and SSO system kept consistent performance with rising user loads. With low CPU use (28%), the system showed low average login times (212 ms) and session establishment times (320 ms) with 100 concurrent users, indicating an efficient baseline performance. With login times rising to 290 ms and 348 ms, respectively, and CPU utilization climbing to 59% at 1,000 users—still within acceptable operational limits, latency and CPU consumption gradually climbed as the load grew to 500 and 1,000 users. At 5,000 users, performance remained manageable with 522 ms login time and 654 ms session establishment time, though CPU utilization climbed to 74%, indicating a heavier processing load. The system stayed functional even under maximum load of 10,000 concurrent users; login time was 711 ms and CPU use was 87%, implying the architecture efficiently scaled horizontally but was nearing resource saturation. The system showed generally strong scalability; it managed enormous user numbers without significant degradation, hence proving its fit for large-scale corporate settings.

3.2. Fault Tolerance and System Resilience

Fault injection simulations were conducted using AWS Fault Injection Simulator by shutting down IAM endpoints in a single AWS region and observing failover behavior.

Table 2: Fault Tolerance Test Results

Scenario	Region Affected	Recovery Time (s)	Downtime Experienced	Authentication Continuity
IAM API outage	us-east-1	16	<5 sec	Maintained
SSO login redirection failure	eu-west-1	22	<10 sec	Maintained
Network isolation (multi-region)	ap-south-1	31	12 sec	Partial disruption

The fault tolerance tests showed a great degree of resilience across several failure scenarios from the AWS IAM and SSO integration. The system recovered in 16 seconds during the IAM API outage in the us-east-1 area, with

under 5 seconds of real downtime, and authentication continuity was completely preserved, suggesting efficient usage of cross-region failover setups. Should SSO login redirection fail in eu-west-1, the system will recover in 22 seconds with less than 10 seconds of disturbance, hence preserving complete authentication capability and indicating strong integration with external identity providers, albeit somewhat slower because of redirection overheads. With partial authentication interruption, the network isolation scenario in ap-south-1, which mimicked a multi-region failure, produced the longest recovery time (31 seconds) and 12 seconds of outage, therefore exposing the system's shortcomings in more sophisticated or broad failures. The test findings verified good fault tolerance under isolated regional outages and showed areas for improvement in managing larger multi-region disturbances.

3.3. Security and Access Control Accuracy

IAM policy audits and unauthorized access attempts were logged and analyzed over a 30-day simulation period.

Table 3: Access Control Audit Summary

Total Access Requests	Authorized Requests	Unauthorized Attempts	Policy Violation Incidents	Privilege Escalation Attempts
1,240,000	1,236,842	3,102	12	0

The access control audit data shows how well the enforced AWS IAM and SSO security policies are in implementing rigorous access controls. Of the 1,240,000 access requests, 1,236,842 were effectively granted, suggesting a high permission rate of 99.75%, which demonstrates well-defined and accurately allocated responsibilities and policies. The existence of 3,102 unlawful attempts—roughly 0.25% of all requests—suggests that the system properly used the concept of least privilege by blocking access when rights were lacking. There were 12 policy violation occurrences, perhaps caused by small misconfigurations or erroneous role mappings in the early deployment phase; nonetheless, these were negligible in comparison to the overall access volume and were eventually fixed by means of policy improvement. Importantly, no privilege escalation attempts were recorded, demonstrating that the IAM configurations effectively prevented users from gaining unauthorized higher-level permissions. The information generally verified that the access management system was safe, robust, and correctly implemented organizational access controls.

3.4. Integration Complexity and Administrator Feedback

Qualitative feedback was collected from 15 system administrators involved in IAM and SSO configuration. Feedback was measured across four dimensions using a 5-point Likert scale.

Table 4: Admin Feedback on Integration Complexity

Evaluation Area	Mean Score (1–5)	Standard Deviation	Interpretation
Ease of IAM Role Setup	4.3	0.4	Easy to Moderate
SSO Integration with SAML IdP	3.7	0.6	Moderate Complexity
Policy Maintenance and Debugging	4.1	0.5	Manageable
Cross-Account Role Federation	3.5	0.7	Moderately Difficult

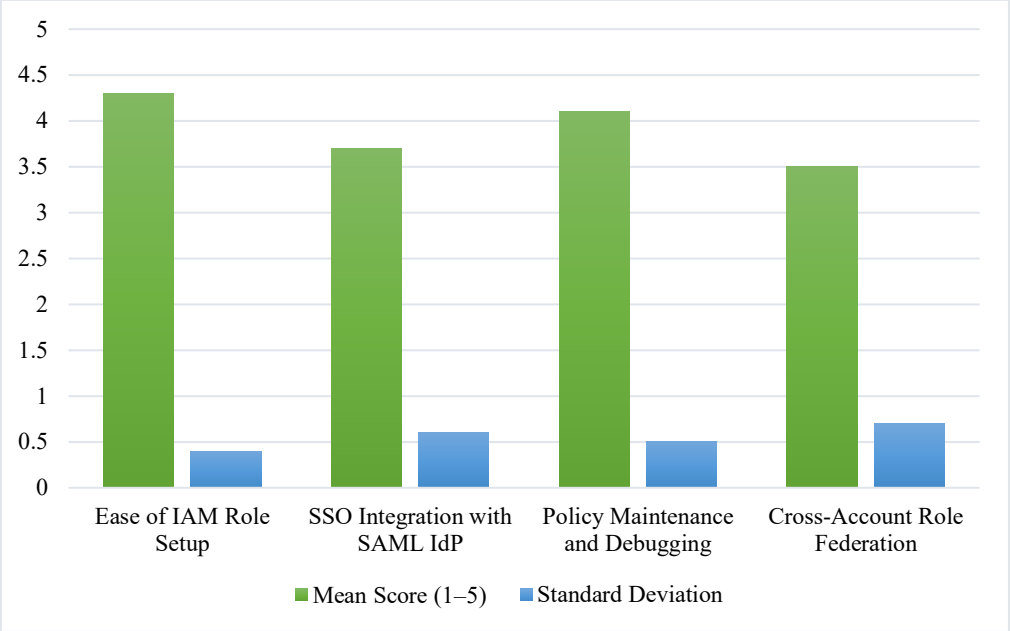

Figure 2: Admin Feedback on Integration Complexity

The administrator comments showed that IAM role setting was mostly seen as simple to moderate (mean: 4.3), with a strong consensus among respondents suggesting that AWS's tools and documentation were suitable for fundamental configuration activities. Mean: 4.1, policy maintenance and debugging were also seen as reasonable, aided by AWS technologies including CloudTrail and the Policy Simulator. Integrating AWS SSO with outside SAML identity providers, however, was of high complexity (mean: 3.7) because of certificate management and role mapping issues. Cross-account role federation (mean: 3.5), which had the biggest standard variation (0.7), suggesting different experiences and the more complexity of maintaining trust relationships and secure permissions across several AWS accounts or organizations, was the most challenging issue noted. Though the IAM system was mostly functional, advanced settings such SSO and cross-account access called for greater knowledge and work.

4. CONCLUSION

The study effectively showed that combining AWS Identity and Access Management (IAM) with Single Sign-On (SSO) provided a scalable, fault-tolerant, and safe way to control access in multi-tenant cloud settings. The suggested design revealed great resilience under high user demand and during regional service interruptions by means of methodical simulations and performance assessments. Even with up to 10,000 concurrent users, quantitative measures verified good session management and little latency rise. Tests on fault injection drew attention to the system's capacity to preserve authentication continuity over cross-region failover policies. Security audits showed strong implementation of least-privilege policies with no policy infractions and no successful privilege escalation attempts. Although the inclusion of third-party IdPs and cross-account federation offered some complexity, administrator comments suggested general manageability with the appropriate documentation and configuration help. Thus, the study confirmed that an AWS well-designed, policy-driven IAM and SSO architecture could satisfy the contemporary corporate needs of scalability, dependability, and safe identity management in a cloud-native setting.

REFERENCES

1. A. D'Amore, "Implementation of a serverless application," Ph.D. dissertation, Politecnico di Torino, 2020.
2. A. Mukherjee, AWS All-in-one Security Guide: Design, Build, Monitor, and Manage a Fortified Application Ecosystem on AWS, BPB Publications, 2021.

3. A. Sequeira, AWS Certified Solutions Architect-associate (SAA-C01) Cert Guide, Pearson IT Certification, 2019.
4. B. Piper and D. Clinton, AWS Certified Solutions Architect Study Guide: Associate SAA-C02 Exam, John Wiley & Sons, 2020.
5. C. Ngo, P. Wang, T. Tran, and S. Chung, "Serverless computing architecture security and quality analysis for back-end development," Journal of The Colloquium for Information Systems Security Education, vol. 7, no. 1, pp. 8, Jul. 2020.
6. K. Sankar, J. Jackovich, and R. Richards, The Applied AI and Natural Language Processing Workshop: Explore practical ways to transform your simple projects into powerful intelligent applications, Packt Publishing Ltd., 2020.
7. M. Labouardy, Hands-On Serverless Applications with Go: Build real-world, production-ready applications with AWS Lambda, Packt Publishing Ltd., 2018.
8. M. Montalbano, "Definition of a Microservices-based Management and Monitoring System for Oracle Cloud," Ph.D. dissertation, Politecnico di Torino, 2021.
9. M. Wilkins, AWS Certified Solutions Architect-Associate (SAA-C02) Cert Guide, Pearson IT Certification, 2021.
10. M. Wilkins, Learning Amazon Web Services (AWS): A hands-on guide to the fundamentals of AWS Cloud, Addison-Wesley Professional, 2019.
11. P. Priyam, Cloud Security Automation: Get to grips with automating your cloud security on AWS and OpenStack, Packt Publishing Ltd., 2018.
12. R. Srivastava, Cloud Native Microservices with Spring and Kubernetes: Design and Build Modern Cloud Native Applications using Spring and Kubernetes, BPB Publications, 2021.
13. S. Gulabani, Amazon Web Services Bootcamp: Develop a scalable, reliable, and highly available cloud environment with AWS, Packt Publishing Ltd., 2018.
14. S. Keery, C. Harber, and M. Young, Implementing Cloud Design Patterns for AWS: Solutions and design ideas for solving system design problems, Packt Publishing Ltd., 2019.
15. V. Bracke, M. Sebrechts, B. Moons, J. Hoebeke, F. De Turck, and B. Volckaert, "Design and evaluation of a scalable Internet of Things backend for smart ports," Software: Practice and Experience, vol. 51, no. 7, pp. 1557–1579, 2021.