International Journal of
Computing and
Engineering
(IJCE)

Optimizing Legacy Systems for Cloud Migration: Patterns and Pitfalls in AWS Transition

CARI Journals

International Journal of Computing and Engineering
ISSN 2958-7425 (online)
Vol. 4, Issue No. 4, pp. 6 - 16, 2023
www.carijournals.org

Optimizing Legacy Systems for Cloud Migration: Patterns and Pitfalls in AWS Transition

Sandeep Patil
Shell
https://orcid.org/0009-0003-4504-543X

Abstract

Legacy systems, while foundational to many enterprises, increasingly hinder agility, scalability, and cost-efficiency in today's digital landscape. Cloud migration particularly to Amazon Web Services (AWS) offers a path to modernization, but the journey is fraught with architectural, operational, and organizational challenges. This paper explores the strategic optimization of legacy systems during AWS migration, emphasizing established migration patterns such as rehosting, replatforming, and refactoring. Drawing from real-world implementations, I analyze common pitfalls including data integrity issues, cost overruns, security vulnerabilities, and resistance to change. I also present best practices for pre-migration assessment, infrastructure automation, and post-migration performance tuning. Special attention is given to AWS-native tools and services that support a secure, efficient transition. The article provides actionable insights for IT leaders, architects, and decision-makers. This research aims to demystify the AWS migration process and offer a roadmap for organizations seeking to transform legacy systems into scalable, cloud-native solutions while minimizing disruption and maximizing return on investment.

Keywords: *Cloud Migration, Legacy Systems, Amazon Web Services (AWS), Rehosting, Refactoring, Cloud Optimization, Hybrid Cloud*

1. INTRODUCTION

In the digital landscape, organizations increasingly face pressure to modernize their legacy systems to remain competitive, secure, and agile. Legacy systems often characterized by monolithic architectures, outdated technologies, and high maintenance costs can significantly hinder innovation and scalability. Cloud migration, particularly to Amazon Web Services (AWS), offers a viable solution by providing scalable infrastructure, reduced operational costs, and improved system reliability [1]. AWS, as a leading cloud provider, offers a comprehensive suite of services and tools tailored for enterprise-scale migrations. The migration process is not without complexity. It involves not only technical restructuring but also strategic alignment across business units, risk mitigation, and compliance assurance. Migrating legacy workloads to the cloud introduces numerous challenges, including data dependency issues, insufficient automation, and a lack of cloud-native skills within organizations [2].

2. UNDERSTANDING LEGACY SYSTEMS

Legacy systems are long-standing information systems that continue to fulfill core business functions but rely on outdated hardware, software platforms, or programming languages. These systems often lack compatibility with modern technologies, present scalability limitations, and incur high operational costs. Despite these shortcomings, they are typically mission-critical, housing essential business logic, data, and processes built over decades [4]. The enduring presence of legacy systems across industries such as finance, healthcare, and government is largely due to their reliability and the substantial cost and risk involved in replacing them. Over time, these systems become increasingly fragile, harder to maintain, and resistant to integration with newer digital services. Common characteristics include monolithic architectures, tightly coupled components, and obsolete programming environments such as COBOL or mainframe systems [5].

A key challenge in dealing with legacy systems is technical debt the accumulated consequences of earlier design or implementation decisions that may no longer be optimal. Technical debt inhibits agility and complicates system enhancements or integration efforts [6]. Legacy systems frequently lack comprehensive documentation, making system understanding and modernization more difficult. Security vulnerabilities also proliferate in aging systems that are no longer supported by vendors. This poses compliance risks, particularly for organizations subject to evolving regulatory standards. The diminishing pool of skilled personnel capable of maintaining these platforms further exacerbates operational risk [7]. Recognizing the limitations and dependencies of legacy systems is essential before embarking on a cloud migration strategy. A comprehensive assessment of system architecture, business value, and risk exposure lays the groundwork for selecting an appropriate migration path on AWS.

3. OVERVIEW OF AWS CLOUD PLATFORM

Amazon Web Services (AWS) is one of the most widely adopted cloud computing platforms, offering over 200 fully featured services spanning compute, storage, database, networking,

machine learning, and security. Its global infrastructure is built for scalability, availability, and operational resilience, making it a preferred platform for enterprises migrating legacy systems to the cloud [8]. At the core of AWS are foundational services such as Amazon Elastic Compute Cloud (EC2) for scalable virtual servers, Amazon Simple Storage Service (S3) for object storage, and Amazon Relational Database Service (RDS) for managed databases like MySQL, PostgreSQL, and Oracle. These services enable organizations to replace or augment on-premises infrastructure without the need for physical hardware investments [9].

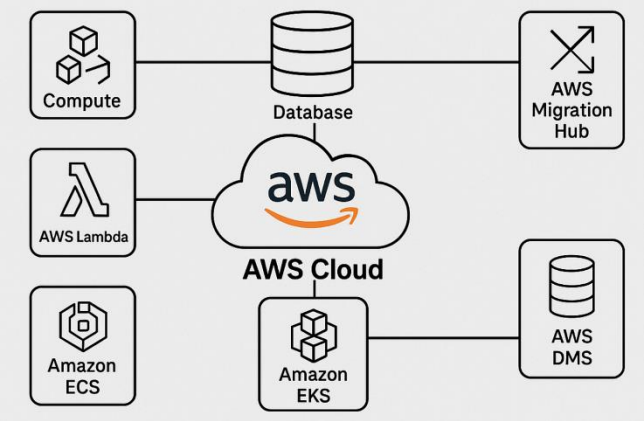

Figure1. Overview of AWS Cloud Platform

AWS also offers specialized tools to facilitate migration. The AWS Migration Hub provides a unified dashboard to track the progress of application migrations across AWS services. AWS Application Discovery Service helps identify server dependencies and usage patterns critical for planning large-scale migrations. For data centric transitions, tools like AWS Database Migration Service (DMS) and Snowball support seamless data transfer and transformation [10]. The flexibility of AWS extends to serverless computing through AWS Lambda, allowing organizations to refactor monolithic applications into microservices with minimal infrastructure overhead. AWS supports containerized workloads using Amazon ECS and EKS, enabling smooth modernization of legacy applications [11].

AWS embeds robust security and compliance features, including Identity and Access Management (IAM), encryption at rest and in transit, and a shared responsibility model. These capabilities are essential for organizations transitioning mission critical legacy systems while adhering to industry regulations. Understanding the capabilities and service ecosystem of AWS is a prerequisite for designing optimized cloud architectures that align with business goals and technical constraints.

4. Migration Patterns and Strategies

When migrating legacy systems to the cloud, choosing an appropriate migration strategy is critical for minimizing risk, managing costs, and achieving long-term operational benefits. Amazon Web Services (AWS) promotes a widely adopted framework known as the “7 R’s of Migration” a

taxonomy of seven strategic approaches that guide enterprises through different modernization paths: Rehost, Replatform, Repurchase, Refactor, Retire, Retain, and Relocate [12].

Rehosting (Lift and Shift) is often the fastest method, involving minimal changes by moving applications from on-premises infrastructure directly to the cloud EC2 instances. It's ideal for organizations seeking rapid migration with limited upfront investment, though it may not leverage cloud-native benefits [13].

Replatforming (Lift, Tinker, and Shift) modifies components to better suit the cloud environment without redesigning the application architecture. An example is migrating a database from on-premises Oracle to Amazon RDS with minimal changes [14].

Repurchasing involves replacing legacy systems with commercial SaaS solutions, offering reduced maintenance overhead but often requiring workflow adaptation. Refactoring (Rearchitecting) entails significant changes to application architecture converting a monolithic system to microservices using AWS Lambda or containers. Though resource-intensive, this unlocks the greatest scalability and agility [15]. Retire involves decommissioning obsolete applications, while Retain applies to systems that must remain on-premises temporarily due to compliance or latency concerns. Relocate, a more recent addition, enables organizations to move large-scale VMware workloads to AWS without refactoring, preserving configurations and dependencies [16].

Selecting the appropriate strategy requires a careful evaluation of system dependencies, business value, technical feasibility, and cloud readiness. Hybrid strategies, where different workloads follow different migration paths, are increasingly common in large enterprises. Tools like the AWS Migration Readiness Assessment (MRA) and Application Discovery Service support informed decision-making during this planning phase [17].

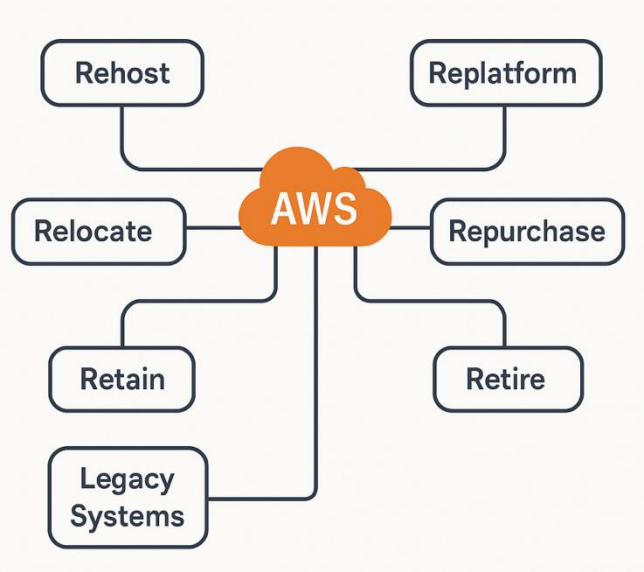

Figure 2. Migration Patterns and Strategies

The success of a migration strategy lies in its alignment with organizational goals, user expectations, and post-migration operational models. Enterprises must balance short-term execution speed with long-term optimization and resilience.

5. Common Pitfalls and Challenges

Migrating legacy systems to AWS, while promising in terms of scalability and modernization, is fraught with technical, operational, and organizational challenges. These pitfalls can lead to cost overruns, project delays, performance degradation, or even migration failure if not addressed proactively. One of the most frequent issues is underestimating system complexity. Legacy applications often have poorly documented architectures and hard-coded dependencies, which hinder automated discovery and make accurate dependency mapping difficult [18]. This increases the risk of application failures post-migration. Another common challenge is data migration complexity, particularly with large volumes of transactional or relational data. Inconsistencies between data schemas, lack of synchronization mechanisms, or downtime constraints can result in data loss or corruption [19]. These problems are exacerbated in environments with real-time data flows and tight service-level agreements (SLAs).

Security and compliance gaps often emerge when legacy controls are not mapped adequately to cloud-native counterparts. Organizations may neglect critical configurations like Identity and Access Management (IAM), network segmentation, and encryption policies during rapid rehosting or replatforming efforts [20]. Cost mismanagement is a well-documented pitfall. While cloud models offer pay-as-you-go flexibility, failure to right-size compute instances or use reserved

pricing models can result in significantly higher-than-expected expenditures [21]. Organizations frequently overlook post-migration optimization, leading to inefficient resource utilization.

From an organizational perspective, skills shortages and resistance to change are major obstacles. Many enterprises lack in-house expertise in AWS services and DevOps practices, which limits their ability to fully leverage the platform. Moreover, without proper change management, teams may continue operating as if in an on-premises environment, undermining the value of the cloud migration [22]. Lack of a phased migration plan and governance model often leads to fragmentation and inconsistent adoption of best practices across business units. A successful migration requires not just technical reconfiguration, but also process realignment, stakeholder buy-in, and continuous monitoring.

6. Optimization Techniques

Optimizing legacy systems during and after migration to AWS is crucial for achieving long-term performance, cost-efficiency, and maintainability. A successful optimization strategy involves not only selecting the right migration pattern but also applying technical best practices at each stage of the cloud adoption lifecycle pre-migration, during transition, and post-migration. Pre-migration assessment is foundational. Organizations must perform detailed dependency mapping, workload analysis, and cloud readiness evaluations using tools such as AWS Migration Readiness Assessment (MRA) and AWS Application Discovery Service [23]. This helps in identifying performance bottlenecks and underutilized assets, enabling informed decisions on right-sizing and refactoring.

During the transition, Infrastructure as Code (IaC) via tools like AWS CloudFormation and Terraform can be used to automate resource provisioning. This approach improves repeatability, reduces manual errors, and supports version control in infrastructure deployment [24]. Post-migration, optimization must focus on performance tuning and cost governance. Services such as AWS Compute Optimizer, Trusted Advisor, and Cost Explorer help in identifying underutilized resources and recommending instance type changes or autoscaling configurations [25]. Implementing autoscaling groups, enabling Elastic Load Balancing (ELB), and leveraging Amazon CloudWatch metrics are vital for maintaining application responsiveness under varying workloads.

Modernization through containerization and serverless computing allows legacy applications to take advantage of cloud-native benefits. Migrating workloads to Amazon ECS, EKS, or AWS Lambda reduces overhead and enhances scalability, especially for modular or event-driven architectures [26]. Another essential practice is the use of continuous monitoring and observability frameworks. Integrating services such as AWS CloudTrail, CloudWatch, and X-Ray enables real-time visibility into application behavior, latency patterns, and security compliance, facilitating early anomaly detection and rapid response [27].

Cost optimization is an ongoing process. Leveraging reserved instances, spot instances, and Savings Plans can dramatically lower operating expenses. Effective tagging strategies and account-level budgets ensure clear chargebacks and accountability across business units [28]. By applying these optimization techniques holistically, organizations can ensure that their AWS migration not only meets immediate technical goals but also aligns with broader strategic objectives for agility, security, and operational efficiency.

7. Security and Compliance Considerations

Security and compliance are central to any cloud migration initiative, particularly when transitioning legacy systems that may lack robust controls or alignment with modern standards. As organizations move to AWS, they must adopt a security-first mindset to ensure data confidentiality, integrity, and availability, while also maintaining adherence to regulatory and industry-specific compliance mandates.

A foundational principle in AWS is the Shared Responsibility Model, where AWS secures the infrastructure hardware, networking, facilities, while customers are responsible for securing their applications, data, and access configurations [29]. Failure to fully understand and implement this model can expose migrated systems to vulnerabilities.

Identity and Access Management (IAM) is a critical area. Organizations should enforce least-privilege access, apply multi-factor authentication (MFA), and use role-based access control (RBAC) to limit and audit user permissions. AWS IAM, along with AWS Organizations and Service Control Policies (SCPs), provides granular access management at both account and service levels [30].

Data protection must be considered throughout the migration process. Legacy data often lacks encryption or version control. AWS offers in-transit and at-rest encryption through services like AWS Key Management Service (KMS) and CloudHSM. Proper encryption key lifecycle management is crucial to maintaining data confidentiality and compliance [31].

Logging, monitoring, and auditing must be integrated early in the migration lifecycle. AWS CloudTrail, AWS Config, and Amazon GuardDuty enable continuous tracking of configuration changes, policy violations, and threats. These services help demonstrate compliance and provide forensic capabilities in case of security incidents [32].

Network security is another key pillar. Legacy systems often rely on flat, perimeter-based security models, whereas AWS promotes defense-in-depth using VPCs, subnet isolation, network ACLs, and security groups to segment and control traffic. AWS Web Application Firewall (WAF) and Shield Advanced provide additional layers of protection against DDoS and application-level attacks [33].

8. Future Directions

As cloud technologies and enterprise demands evolve, the future of legacy system migration to AWS will be shaped by innovations that emphasize automation, intelligence, and continuous modernization. Several emerging trends are expected to significantly influence the strategies, tools, and success metrics used in cloud transformation efforts.

One key direction is the increased adoption of AI/ML driven migration tooling. Amazon and third-party providers are developing machine learning powered platforms that assist in application discovery, dependency mapping, cost forecasting, and risk assessment. These tools reduce human error and accelerate decision making during complex migrations. Intelligent orchestration platforms can further automate rollback procedures, configuration adjustments, and performance tuning based on real time analytics.

Serverless and container-first architectures will continue to gain prominence. Organizations are increasingly bypassing traditional virtual machines in favor of AWS Lambda, ECS, and EKS to achieve greater scalability and cost-efficiency. Migration strategies are expected to evolve from lift-and-shift to cloud-native transformations, wherein legacy components are refactored directly into microservices or functions-as-a-service (FaaS). Edge computing and hybrid cloud will also influence migration strategies, especially in industries requiring low-latency data processing or regulatory data residency. AWS services like Snowcone, Wavelength, and Outposts will play a larger role in enabling hybrid and edge-native architectures.

9. Conclusion

Migrating legacy systems to AWS represents a transformative opportunity for organizations seeking to enhance agility, scalability, and operational efficiency. This process is complex and demands a strategic approach that balances technical feasibility with business objectives. By examining AWS migration patterns such as rehosting, replatforming, and refactoring and identifying common pitfalls, this article provides a roadmap to mitigate risk and maximize return on investment. Key challenges such as system complexity, data integration, compliance, and cost management can be overcome through thorough pre-migration assessments, automation via Infrastructure as Code, and continuous optimization using AWS-native tools. Security and governance must be embedded from the outset, ensuring long-term resilience and regulatory alignment.

Looking ahead, advances in AI-driven migration tools, serverless computing, and hybrid cloud solutions will reshape how legacy systems are modernized. Organizations must shift from one-time migration efforts to a continuous modernization mindset, leveraging DevSecOps and policy-as-code frameworks. Successful legacy system migration is not solely a technical exercise but a catalyst for enterprise-wide digital transformation. With the right patterns, tools, and governance in place, AWS offers a powerful platform to evolve legacy infrastructure into scalable, secure, and cloud-native ecosystems.

REFERENCES

[1] A. Khajeh-Hosseini, D. Greenwood, J. W. Smith, and I. Sommerville, "The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise," Software: Practice and Experience, vol. 42, no. 4, pp. 447–465, 2012.

[2] G. Lewis, "The Role of Architecture Evaluation in System Migration," in Proc. 2011 Joint Working IEEE/IFIP Conf. on Software Architecture (WICSA), pp. 284–287.

[3] AWS, "Migration Strategies – The 7 R's," Amazon Web Services,[Online].Available: [https://aws.amazon.com/blogs/enterprise-strategy/migration-strategies-the-7-rs/]

[4] D. S. Linthicum, Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide, Addison-Wesley, 2009.

[5] M. Richards, "Software Architecture Patterns," O'Reilly Media, 2015.

[6] P. Avgeriou, P. Kruchten, I. Ozkaya, and C. Seaman, "Managing Technical Debt in Software Engineering," Dagstuhl Reports, vol. 6, no. 4, pp. 110–138, 2016.

[7] R. L. Glass, "Facts and Fallacies of Software Engineering," Addison-Wesley, 2003.

[8] J. Varia and S. Mathew, "Overview of Amazon Web Services," AWS Whitepaper, Amazon Web Services, Jan. 2021.[Online].Available: [https://docs.aws.amazon.com/whitepapers/latest/aws-overview/aws-overview.pdf]

[9] B. Sosinsky, Cloud Computing Bible, Wiley Publishing, 2011.

[10] Amazon Web Services, "Migration Tools and Services," [Online]. Available: [https://aws.amazon.com/migration-tools/]

[11] M. Haines, "Serverless Architectures with AWS Lambda," O'Reilly Media, 2017.

[12] Amazon Web Services, "Migration Strategies – The 7 R's,"[Online].Available: [https://aws.amazon.com/blogs/enterprise-strategy/migration-strategies-the-7-rs/]

[13] B. Golden, Amazon Web Services for Dummies, 2nd ed., Wiley, 2019.

[14] AWS, "Migration Whitepaper: Best Practices for Cloud Migrations," Amazon Web Services, 2020. [Online]. Available: [https://d1.awsstatic.com/whitepapers/AWS_Cloud_Migration_Strategy.pdf]

[15] M. Villamizar et al., "Evaluating the Monolithic and the Microservice Architecture Pattern to Deploy Web Applications in the Cloud," in Proc. 10th Computing Colombian Conf. (10CCC), 2015, pp. 583–590.

[16] VMware, "VMware Cloud on AWS:Technical Overview,"2022.[Online].Available: [https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vmc/vmware-cloud-aws-technical-overview.pdf]

[17] D. S. Linthicum, "Cloud Migration Success Factors," IEEE Cloud Computing, vol. 5, no. 1, pp. 66–70, Jan./Feb. 2018.

[18] R. T. Fielding and R. N. Taylor, "Principled Design of the Modern Web Architecture," ACM Trans. Internet Technol., vol. 2, no. 2, pp. 115–150, May 2002.

[19] A. Bouguettaya, Q. Yu, X. Liu, and H. Dong, "Web Service Provisioning: State-of-the-Art and Research Directions," IEEE Internet Computing, vol. 18, no. 5, pp. 46–56, Sept.–Oct. 2014.

[20] AWS, "Security Best Practices for AWS Migrations," AWS Whitepaper, 2022. [Online]. Available: [https://d1.awsstatic.com/whitepapers/aws-security-best-practices.pdf]

[21] B. Jennings and R. Stadler, "Resource Management in Clouds: Survey and Research Challenges," J. Netw. Syst. Manage., vol. 23, no. 3, pp. 567–619, July 2015.

[22] M. Khajeh-Hosseini, I. Sommerville, J. Bogaerts, and P. Teregowda, "Decision Support Tools for Cloud Migration in the Enterprise," in Proc. IEEE Int. Conf. Cloud Computing (CLOUD), 2011, pp. 541–548.

[23] AWS, "Migration Readiness Assessment Guide," AWS Whitepaper,2021.[Online].Available: [https://d1.awsstatic.com/whitepapers/migration-readiness-assessment.pdf](

[24] P. Sharma, A. Jindal, and J. Singh, "Infrastructure as Code: Implementation and Challenges," in Proc. Int. Conf. Smart Electronics and Communication (ICOSEC), 2020, pp. 1–5.

[25] A. Barker, B. Varghese, and L. Thai, "Cloud Services Benchmarking for Performance," IEEE Cloud Computing, vol. 3, no. 1, pp. 32–40, Jan.–Feb. 2016.

[26] J. Spillner, "Transcending Cloud Elasticity: Service Auto-Scaling at Application Level," in Proc. IEEE World Congr. Services, 2017, pp. 1–6.

[27] T. Chen, D. S. Rosenblum, and M. Zhang, "Towards Monitoring and Detecting QoS Violations in Cloud Computing," in Proc. IEEE/ACM Int. Symp. Cluster, Cloud and Grid Computing (CCGrid), 2014, pp. 327–336.

[28] AWS, "AWS Cost Optimization: Best Practices," AWS Whitepaper,2022.[Online].Available: [https://d1.awsstatic.com/whitepapers/aws-cost-optimization.pdf]

[29] AWS, "AWS Shared Responsibility Model," AWS Whitepaper,2022.[Online].Available: [https://d1.awsstatic.com/whitepapers/aws-shared-responsibility-model.pdf]

[30] A. Nadalin, E. Bertino, and A. J. Lee, "Role-Based Access Control," in Proc. 5th ACM Symp. Access Control Models and Technologies, 2000, pp. 47–63.

[31] AWS, "AWS Key Management Service Best Practices," AWS Whitepaper,2021. [Online]. Available: [https://docs.aws.amazon.com/kms/latest/developerguide/best-practices.html]

[32] A. Shabtai, Y. Elovici, and L. Rokach, "A Survey of Data Leakage Detection and Prevention Solutions," Springer Data Mining and Knowledge Discovery, vol. 14, no. 5, pp. 1–20, 2012.

[33] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in Cloud Computing: Opportunities and Challenges," Information Sciences, vol. 305, pp. 357–383, June 2015.