ESP-IJACT
ESP International Journal of Advancements in Computational Technology
ISSN: 2583-8628 / Volume 1 Issue 3 December 2023 / Page No: 97-103
Paper Id: IJACT-V1I3P110 / Doi: 10.56472/25838628/IJACT-V1I3P110

Original Article

Cloud Migration Strategies: Best Practices for Moving Legacy Applications to the Cloud

Nikhil Bhagat
Sr. Technical Account Manager – Network Specialist, Independent Scholar, Network Engineering, USA.

Received Date: 28 October 2023 Revised Date: 27 November 2023 Accepted Date: 21 December 2023

Abstract: *Migration to the cloud is a significant move for enterprises seeking to transform their IT platforms and remain relevant in a digital marketplace. Existing applications that are based on legacy frameworks are difficult to scale, adjust and maintain affordably. By moving these applications to the cloud, companies can benefit from modern cloud capabilities like on-demand scaling, faster performance and improved security. This paper presents best practices for migrating legacy applications to the cloud, covering various cloud migration techniques such as rehosting, replatforming, and refactoring. It also elaborates on the key elements involved in creating a successful cloud migration roadmap, from assessing the current state of deployment to selecting applications and ensuring secure deployment. Further, the paper goes into design considerations like performance optimization, resilience, and cost to ensure migrated applications are effectively running in the cloud. When implementing a systematic cloud migration while considering both the technical and business implications, businesses will not need to worry about the transition and can fully reap the benefits of cloud computing.*

Keywords: *Cloud Migration Strategies Rehosting, Replatforming, Refactoring, Repurchasing, Best Practices.*

I. INTRODUCTION

With the explosive development of cloud computing, organizations have been able to manage IT resources more effectively, using scalable, cost-effective, and adaptive solutions that were not possible with conventional infrastructure. This is why, companies in every sector have been proactively moving applications, services, and workloads to the cloud for competitive advantages [1]. Among these, the migration of legacy applications, which have used legacy technology for years, run on old platforms, and are typically on-premises, represents a significant modernization opportunity.

Legacy applications, though an integral part of business operations, always pose certain challenges. They're typically costly to maintain, hard to scale and possibly too inflexible to evolve with the changing requirements of today's business [2]. Also, they tend to be unfit for new technologies and development methodologies, which results in waste of resources and slowed innovation. By migrating these applications to the cloud, these constraints are addressed, and enterprises can also benefit from the advanced features of cloud infrastructures, such as faster performance, global availability, enhanced security, and automated scalability [3].

However, migrating legacy applications into the cloud can be a challenging process. It requires significant planning, a decision on the migration method, and knowledge of the application's architecture and dependencies. Businesses should also think about the unique properties of their applications and the business requirements when they are planning to use rehosting, replatforming or refactoring strategy [4].

The paper describes best practices for moving legacy applications to the cloud, including the benefits of cloud migration, migration approaches, and ways to design a successful cloud migration strategy. It also covers the design considerations a business needs to consider in the migration effort to support legacy applications within the cloud. By using a planned and well educated strategy, companies can successfully transition and fully reap the advantages of cloud technologies.

II. BENEFITS OF MIGRATING TO THE CLOUD

Moving legacy apps to the cloud brings multiple advantages that companies can utilize to improve efficiency and innovation:

A. Effective Cost Management

A key factor for cloud migration is the chance to lower infrastructure and management costs. Traditional data centers need lots of initial capital, and are incredibly costly to maintain. By relocating to the cloud, companies can shift from a capital expenditure (CapEx) model to an Operating Expenditure (OpEx) model in which they only pay for what they consume [5]. This will be easier to budget and minimize the cost of costly hardware upgrades.

B. Provides Scalability

Cloud services are nearly limitless in scale, meaning companies can scale as needed [6]. This is especially useful for applications that are dependent on heavy traffic. The elastic nature of the cloud ensures that enterprises can scale up during the peak periods and lower their capacity during the off-peak periods, which helps to optimize usage and minimize costs.

C. High Availability and Reliability

Cloud providers offer High availability, disaster recovery, fault tolerance, and scalability due to their efficient architectures. When a business migrates legacy applications to the cloud, it will benefit from these features and achieve higher uptime and application performance [7]. Many cloud providers also offer global availability zones for deploying applications closer to users in order to minimize latency.

D. Improved Security

Modern cloud services provide a range of advanced security measures such as encryption, identity and access management (IAM), and industry standard compliance (e.g., GDPR, HIPAA) [8]. By moving legacy applications to the cloud, companies gain a boost in security with these options, and will not be vulnerable to data leaks or cyberattacks [9].

E. Fuels Innovation

Cloud Computing accelerates time-to-market by providing faster release of new features and enhancements [10]. For example, legacy applications use dated pipelines for development and deployment, which hampers innovation in a company. Migrating to the cloud also enables enterprises to embrace modern DevOps, automation, and new technologies such as machine learning, AI, and IoT that enable a culture of innovation.

III. CLOUD MIGRATION STRATEGIES AND BEST PRACTICES

Migrating applications to the cloud can be a challenging process especially when you are dealing with legacy applications [11]. Cloud migration is not a one-size-fits-all initiative and the organization has to develop an appropriate strategy based on the technical requirements, the business objective and the architecture of the existing applications. Over time, the most effective practices for cloud migration have been reduced to a few tips and tricks that should help organizations move forward. These approaches are often referred to as the "6 R's of Cloud Migration". Each of these options - rehosting, replatforming, refactoring, repurchasing, retiring, retaining offer different advantages based on an organization's requirements as well as the applications architecture [12]. This section of the paper delves into cloud migration strategies in terms of their use cases, benefits, limitations, and best practices.

A. Rehosting ("Lift and Shift")

Rehosting (also known as "lift and shift") is the easiest approach to migrate your cloud. Generally, it entails migrating an application from its on-premise deployment to the cloud with little or no change [13]. The intention is to emulate the existing infrastructure and application layer in the cloud without altering the basic functionality but also harnessing the scale and operational effectiveness of the cloud.

a) Rehosting Use Cases

Rehosting is suitable for enterprises that need to migrate in a short time while minimizing the impact on their operations. It tends to be the preferred approach when a company is under a tight deadline, lacks cloud expertise, or has applications that are inherently reliable but expensive to store in their current environments. It's especially appropriate for older applications that weren't explicitly designed for the cloud and businesses that desire a relatively low-risk, low-effort transition to the cloud.

b) Rehosting Benefits

- Speed and Ease of Implementation: Rehosting is the most effective method to migrate applications to the cloud as it makes no code/architectural changes to the application [14].
- Lower Initial Investment: By migrating to the cloud, the organizations can save on the capital and operating expenditure associated with maintaining physical data centers.
- Infrastructure benefits of the Cloud: Even without the architecture, rehosted applications can take advantage of some of the infrastructure benefits of the cloud such as greater resilience, availability zones, and elastic storage [15].

c) Rehosting Limitations

- Limited Optimization: As rehosting entails very little modification, migrated applications might not be able to utilize cloud native capabilities such as auto-scaling, serverless computing or advanced data storage [13].
- Potential Performance Concerns: Applications that were never architected for the cloud environments may experience bottlenecks or inefficiencies when simply rehosted without any changes.

d) Rehosting Best Practices

Rehosting can be an efficient starting point for cloud migration, but it should be part of a phased approach towards a native cloud model [13]. Applications that are rehosted can then be adapted to the cloud by other methods (replatforming or refactoring) to take full advantage of the cloud.

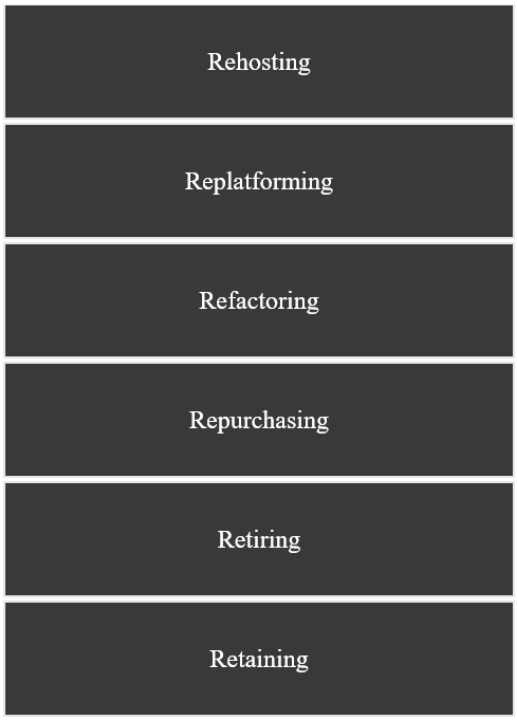

Figure 1: Cloud Migration Strategies

B. Replatforming ("Lift, Tinker, and Shift")

Replatforming is a version of rehosting in which an organization can make minor changes to the application during the migration [16]. The intent here is to re-engineer the application to leverage certain cloud features (managed services, cloud storage, or better databases) without disrupting the core architecture.

a) Replatforming Use Cases

Replatforming is best suited for businesses who need to accelerate and scale applications but are not yet ready or equipped to refactor the entire application [4]. It's ideal for fully functional applications that need cloud-based functionality such as automated scaling, managed services or optimized databases.

b) Replatforming Benefits

- Comparatively lower Cost and Effort: Replatforming is slower and requires more effort than rehosting but is cheaper and time-saving than refactoring [17].
- Optimized for Cloud environments: By exploiting a few cloud-ready offerings such as managed databases or serverless solutions, replatformed applications will gain improved performance and reliability with scale.
- Quicker deployments: Compared to a full refactor, replatforming allows businesses can achieve better performance without needing to make large-scale changes to code or architecture [4].

c) Replatforming Limitations

- Limited Customization: Even though replatforming can be a small part of cloud optimization, it does not provide access to all the cloud native solutions such as microservices or containers [18].
- Continuous Maintenance: Even after a slight adjustment, replatformed apps might need ongoing maintenances to ensure they remain up to speed as the cloud environments evolve.

d) Replatforming Best Practices

When replatforming, an organization needs to see which elements of the application can be easily converted to the cloud. Some common areas include databases, storage, and application hosting platforms [18]. For instance, moving away from self-managed databases to cloud-managed databases can boost the performance and minimize the administrative overhead.

C. Refactoring (Re-Architecting)

Refactoring, or re-architecting, is the most comprehensive cloud migration solution. It is about altering the entire architecture and codebase of the application to leverage cloud native capabilities [4]. This usually involves converting monolithic applications to microservices, serverless computing architectures, modern DevOps and CI/CD (Continuous Integration/Continuous Deployment).

a) Refactoring Use Cases

Refactoring should be utilized by enterprises with a long term plan to modernize their application architecture and make the most of the cloud [19]. It is the perfect match for applications that require significant performance, scalability and maintainability improvements, or that are no longer relevant to the business.

b) Refactoring Benefits

- Native Cloud Optimization: Refactoring enables applications to take full advantage of cloud-native features like serverless computing, auto-scaling, containerization, etc. [20].
- Increased Scalability and Performance: By re-designing apps into smaller, self-contained microservices, enterprises can scale segments of an application independently, increasing their performance efficiency and resource utilization.
- Increased Flexibility and Innovation: Refactoring can enable companies to implement DevOps or other advanced development methodologies that allows new features and enhancements to roll out faster [21].
- Refactoring Limitations
- Higher deployment time: Refactoring involves a lot of time, energy, and skills. It is often the most costly and laborious migration approach.
- Complexity: Re-architecting applications is often not straightforward, especially in large and high-dependencies, mission-critical systems.

c) Refactoring Best Practices

Refactoring should be considered when the return on investment is clearly greater than the overhead, and the company is ready to commit to long-term modernization. It's crucial for organizations to plan strategically and prioritize applications that will benefit most from refactoring. It is recommended to experiment with non-critical applications before extending the critical applications to production.

D. Repurchasing (Replacing)

Repurchasing is a way of transitioning from an existing application to a brand-new, cloud-based application [22]. This typically involves using a Software-as-a-Service (SaaS) solution tailored to the business, taking over the existing application.
Repurchasing Use Cases

Repurchasing is great for old applications that are hard to update and no longer deliver the business value [16]. Businesses might prefer to use a cloud application, instead of migrating or refactoring the legacy application and perform the same functionality at better speed, scale and security.

a) Repurchasing Benefits

- Faster Deployment: SaaS applications can be deployed rapidly so that enterprises can begin using cloud-based tools and features right away.
- Cost Efficiency: Most of the SaaS providers are subscription-based, which saves an organization upfront costs as opposed to maintaining or rewriting older applications.
- Low Maintenance: With SaaS, the maintenance, upgrades and security will be taken care by the service provider, thus allowing internal IT teams to work more productively [23].

b) Repurchasing Limitations

- Data Migration and Integration: Migrating from a legacy application to a new SaaS platform may involve massive data migration and integration with other systems.
- Customization: SaaS applications might lack the flexibility of a legacy application, making them not useful for very specific business requirements [24].

c) Repurchasing Best Practices

When repurchasing, it's imperative to thoroughly analyze the available SaaS options to ensure they meet all of the company's business needs. It's crucial that the migration to the new application/architecture is managed effectively to minimize business interruptions and maintain data integrity throughout the process.

E. Retiring

Retiring strategy is used when an organization no longer needs an application or a service. Instead of moving them to the cloud, they are just retired [25].

a) Retiring Use Cases

Organizations often end up having obsolete applications or redundant components on their infrastructure [26]. They can remove these applications and simplify their architecture while making things less complex and expensive.

b) Retiring Benefits

- Cost Reduction: Retiring idle applications cuts down licensing, maintenance and usage costs.
- Simplified Architecture: Retiring unused applications helps simplify the existing architecture, making management and maintenances easier.
- Retiring Limitations
- Dependency Management: When retiring an application, an organization should ensure that no business operations or other applications depend on it [27].

c) Retiring Best Practices

It is recommended to take an inventory of all applications and systems to evaluate potential candidates for retirement. Make sure to understand and analyze all the architectural dependencies before retiring an application.

F. Retaining

Retaining, or "revisiting", means that applications stay in their current on-premise environment instead of migrating to the cloud [4].

a) Retaining Use Cases

Businesses can maintain very sensitive, strictly regulated legacy applications that are not ready for cloud migration due to cost, complexity, or lack of cloud-readiness [28].

b) Retaining Benefits

- Risk Mitigation: By retaining applications that are not easy to migrate, the risk of business interruption can be minimized.
- Compliance: Organizations may have applications that are required to remain on-premises for compliance or data sovereignty reasons [13].

c) Retaining Limitations

- Hybrid Environments: Keeping some applications on-prem and migrating others to the cloud creates a hybrid environment, which needs careful management [29].

d) Retaining Best Practices

Retaining should be an integral part of a larger hybrid cloud strategy. Businesses should continue to consider whether applications need to remain on-premises or if the applications can at some point be moved to the cloud.

IV. HOW TO CREATE A CLOUD MIGRATION PLAN

Implementing a clearly structured cloud migration strategy is the best way for planning legacy applications migration. The following steps represent the most important phases of a cloud migration plan:

A. Analyze the Current Environment

The initial step in cloud migration strategy is to perform an accurate analysis of existing infrastructure, applications, and dependencies. This includes identifying migration candidates (legacy applications), by testing their performance and understanding their technical architecture [30]. Organizations need to conduct a cost-benefit analysis to determine the financial value of the migration.

B. Set Migration Goals

Ensure the migration goals are clearly defined based on business objectives. Such goals can include cost savings, increased scalability, agility or security. These goals will determine the best possible migration strategy and provide key performance indicators (KPIs) to monitor success [31].

C. Select a Suitable Cloud Provider

It is critical to select the right cloud provider for an organization based on their business requirements. Major cloud providers like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) provide a range of services,

pricing, and support options [32]. Organizations need to assess providers by cost, functionality, security and adherence to industry standards.

D. Prioritize Applications for Migration

There is no need for organizations to migrate all the applications at once. If they prioritize applications based on business necessity, complexity and risk then the migration will be smooth. High Priority Mission-critical and/or cost efficient applications must be moved first, while low priority applications can wait [33].

E. Create a Data Migration and Integration plan

Migration of data is a major concern during cloud migration for legacy apps with big databases. Data migration needs to be planned in such a way that data integrity, security and compliance is not affected during the process [13]. Integration with legacy systems should also be considered, as some legacy applications may require integration with either on-premises systems or other cloud platforms.

F. Perform Security and Compliance Actions

In preparation for cloud migration, companies need to adopt robust security and compliance controls, such as identity and access management, encryption, and ensuring cloud compliance [34]. Cloud providers facilitate many security tools and services that can be utilized to improve the security posture of migrated applications.

G. Test and Optimize

Once applications are successfully migrated to the cloud, organizations should test their performance, security and reliability thoroughly [35]. They should perform load tests, security checks, and performance monitoring to find the bottlenecks or problems. Intense optimization of the application in the cloud will ensure that it runs efficiently.

V. CONCLUSION

Migrating legacy applications to the cloud is a strategic decision to help organizations stay competitive, lower costs, and enhance agility. However, the process of migration has its challenges as legacy applications require thorough planning and preparation to migrate successfully into the cloud. If an organization understands various migration strategies, then it can take a well-informed decision between rehosting for acceleration, refactoring for full cloud-nativeness or adopting a hybrid strategy.

A comprehensive cloud migration strategy plays an important role to mitigate the risk and to make the transition as smooth as possible. The plan would require an assessment of the existing infrastructure, identification of applications that should be migrated first, and security and compliance provisions to safeguard data. Following migration, testing and optimization are as critical to performance and cost reduction as it is to the initial migration.

Performance, latency and resilience are important to design legacy applications to be the best that they can be in the cloud. By prioritizing cloud-native architectures, modularization, and using services of cloud providers, scalability, performance, and reliability are also maximized. Ultimately, cloud migration is a way for companies to modernize their applications, cut cost and increase innovation in the service of long-term growth and digital transformation.

VI. REFERENCES

[1] "Cloud adoption to accelerate IT modernization," Apr. 11, 2018. [Online]. Available: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/cloud-adoption-to-accelerate-it-modernization.

[2] R. K. Mallidi, M. Sharma, and J. Singh, "Legacy Digital Transformation: TCO and ROI Analysis," vol. 12, no. 3, pp. 163-170, Aug. 27, 2021. [Online]. Available: https://doi.org/10.32985/ijeces.12.3.5

[3] R. Rai, S. Mehfuz, and G. Sahoo, "Efficient Migration of Application to Clouds: Analysis and Comparison," vol. 3, no. 3, Feb. 28, 2014. [Online]. Available: https://doi.org/10.7603/s40601-013-0023-z

[4] S. Jain, M. Bhushan, A. Yadav, and D. C. Tiwari, "Incremental Architectural Re-engineering for the Cloud: An Overview," pp. 655-659, Feb. 1, 2016. [Online]. Available: https://doi.org/10.1109/cict.2016.135

[5] A. Greenberg, J. R. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud," Assoc. Comput. Mach., vol. 39, no. 1, pp. 68-73, Dec. 31, 2008. [Online]. Available: https://doi.org/10.1145/1496091.1496103

[6] J. Kozhipurath, "Cloud Service Costing Challenges," pp. 1-6, Oct. 1, 2012. [Online]. Available: https://doi.org/10.1109/ccem.2012.6354604

[7] M. F. Zhani and R. Boutaba, "Survivability and Fault Tolerance in the Cloud," pp. 295-308, Apr. 3, 2015. [Online]. Available: https://doi.org/10.1002/9781119042655.ch12

[8] Y. C. Hsu, Y. S. Wu, T. H. Tsai, Y. Chiu, C. H. Lin, and Z. Chen, "MicroApp: Architecting Web Application for Non-uniform Trustworthiness in Cloud Computing Environment," pp. 97-106, Jun. 1, 2014. [Online]. Available: https://doi.org/10.1109/sere-c.2014.27.

[9] S. S. Bhatia and V. Gupta, "A Multi-Tenant Framework for an Enterprise System on Cloud," vol. 5, no. 2, pp. 740-743, Feb. 5, 2016. [Online]. Available: https://doi.org/10.21275/v5i2.nov153195
[10] R. Nazir, Z. Ahmed, Z. Ahmad, N. N. Shaikh, A. A. Laghari, and K. Kumar, "Cloud Computing Applications," May 21, 2020. [Online]. Available: https://eudl.eu/pdf/10.4108/eai.22-5-2020.164667
[11] S. Y. Nabavi and O. Bushehrian, "An adaptive plan-oriented and continuous software migration to cloud in dynamic enterprises," Wiley, vol. 49, no. 9, pp. 1365-1378, Jun. 27, 2019. [Online]. Available: https://doi.org/10.1002/spe.2725
[12] "Navigating Cloud Migration With These 6 Common Strategies," Dec. 31, 2021. [Online]. Available: https://www.rackwareinc.com/6-common-cloud-migration-strategies
[13] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud Migration Research: A Systematic Review," IEEE, vol. 1, no. 2, pp. 142-157, Jan. 1, 2013. [Online]. Available: https://doi.org/10.1109/tcc.2013.10
[14] S. Li, Y. Zhang, and W. Sun, "Optimal Resource Allocation Model and Algorithm for Elastic Enterprise Applications Migration to the Cloud," Multidiscip. Digit. Publ. Inst., vol. 7, no. 10, p. 909, Oct. 1, 2019. [Online]. Available: https://doi.org/10.3390/math7100909
[15] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," Am. Assoc. Adv. Sci., vol. 53, pp. 07-013, Feb. 10, 2009. [Online]. Available: https://storagemadeeasy.com/files/e4c87f06d452ac24d9bffe15085c4189.pdf
[16] B. C. Tak and C. Tang, "AppCloak: Rapid Migration of Legacy Applications into Cloud," pp. 810-817, Jun. 1, 2014. [Online]. Available: https://doi.org/10.1109/cloud.2014.112
[17] "Migrating to AWS Method 3 - Replatforming, aka Lift-and-Shape," Mar. 26, 2018. [Online]. Available: https://cloudsoft.io/blog/migrating-to-aws-method-3-replatforming-aka-lift-and-shape
[18] V. Nuthula and N. R. Challa, "Cloudifying Apps - A Study of Design and Architectural Considerations for Developing Cloud-enabled Applications with Case Study," pp. 1-7, Oct. 1, 2014. [Online]. Available: https://doi.org/10.1109/ccem.2014.7015487
[19] J. Fritzsch, J. Bogner, A. Zimmermann, and S. Wagner, "From Monolith to Microservices: A Classification of Refactoring Approaches," Springer Science+Business Media, pp. 128-141, Jan. 1, 2019. [Online]. Available: https://doi.org/10.1007/978-3-030-06019-0_10
[20] D. S. Linthicum, "Cloud-Native Applications and Cloud Migration: The Good, the Bad, and the Points Between," IEEE, vol. 4, no. 5, pp. 12-14, Sep. 1, 2017. [Online]. Available: https://doi.org/10.1109/mcc.2017.4250932
[21] T. Offerman, R. Blinde, C. J. Stettina, and J. Visser, "A Study of Adoption and Effects of DevOps Practices," Jun. 19, 2022. [Online]. Available: https://doi.org/10.1109/ice/itmc-iamot55089.2022.10033313
[22] D. Yu et al., "A Practical Architecture of Cloudification of Legacy Applications," vol. 324, pp. 17-24, Jul. 1, 2011. [Online]. Available: https://doi.org/10.1109/services.2011.84
[23] S. Farah, "Cloud computing or software as a service—which makes the most sense for HR?," Wiley, vol. 36, no. 4, pp. 31-37, Jan. 26, 2010. [Online]. Available: https://doi.org/10.1002/ert.20271
[24] Q. H. Vu and R. Asal, "Legacy Application Migration to the Cloud: Practicability and Methodology," pp. 270-277, Jun. 1, 2012. [Online]. Available: https://doi.org/10.1109/services.2012.47
[25] N. K. Singh, "Strategic Operating Model for Enterprise Architecture," RELX Group, Netherlands, Jan. 1, 2018. [Online]. Available: https://doi.org/10.2139/ssrn.3098864
[26] P. V. Beserra, A. Camara, R. M. Ximenes, A. B. Albuquerque, and N. C. Mendonça, "Cloudstep: A step-by-step decision process to support legacy application migration to the cloud," Jan. 1, 2022. [Online]. Available: https://doi.org/10.3390/computers7020012
[27] Y. Yu and S. Madiraju, "Enterprise Application Transformation Strategy and Roadmap Design: A Business Value Driven and IT Supportability Based Approach," Aug. 1, 2014. [Online]. Available: https://doi.org/10.1109/es.2014.37
[28] R. Rai, G. Sahoo, and S. Mehfuz, "Advancements and approaches towards moving from legacy application to cloud," Inderscience Publ., vol. 16, no. 2, p. 114, Jan. 1, 2016. [Online]. Available: https://doi.org/10.1504/ijcnds.2016.074547
[29] O. Pfaff and S. Ries, "Avoiding security integration pitfalls in the cloud," Computer Weekly, vol. 1, no. 3, pp. 41-46, Jan. 1, 2017. [Online]. Available: https://doi.org/10.1109/green.2017.7875380
[30] N. C. Mendonça, "Architectural Options for Cloud Migration," IEEE Comput. Soc., vol. 47, no. 8, pp. 62-66, Aug. 1, 2014. [Online]. Available: https://doi.org/10.1109/mc.2014.203
[31] M. Vuković and J. Hwang, "Cloud migration using automated planning," IEEE Access, vol. 7, pp. 121188-121198, Jan. 1, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2934948
[32] P. J. A. Patel and P. N. Kansara, "Cloud Computing Deployment Models: A Comparative Study," pp. 205-210, May 1, 2013. [Online]. Available: https://doi.org/10.1109/ictac.2013.16
[33] C. J. Ward et al., "Workload Migration into Clouds Challenges, Experiences, Opportunities," pp. 164-171, Jul. 1, 2010. [Online]. Available: https://doi.org/10.1109/cloud.2010.84
[34] Y. Chou and J. Oetting, "Secure System Development for Integrated Cloud Applications," Mar. 2, 2022. [Online]. Available: https://doi.org/10.1109/cybersec.2022.9722897
[35] M. A. Chauhan and M. A. Babar, "Towards Process Support for Migrating Applications to Cloud Computing," Nov. 1, 2012. [Online]. Available: https://doi.org/10.1109/csc.2012.20