# Understanding the “7 Rs”

## Cloud Migration Strategies



— Cloud Migration Strategies

In today’s rapidly evolving technological landscape, businesses are increasingly turning to cloud computing to enhance flexibility, scalability, and innovation.

According to [Gartner](https://www.gartner.com/en/newsroom/press-releases/2023-05-16-gartner-says-4-trends-are-shaping-the-future-of-cloud-data-center-and-edge-infrastructure), 65% of application workloads will be ready for cloud delivery by 2027, up from 45% in 2022.

A report from G2 states that 1 out of 3 organizations will migrate from their legacy services to cloud services by 2025 and 75% of organizations are already keen on building cloud-native applications.

However, migrating to the cloud involves careful planning and execution to ensure a seamless transition while maximizing the benefits of the new environment.

As organizations contemplate migrating their applications, data, and infrastructure to the cloud, they are confronted with various challenges and considerations. A well-defined migration strategy becomes paramount to effectively navigate this complex journey.

From assessing existing assets to determining the most suitable migration approach, this article explores the “7 Rs” framework and aims to equip readers with the knowledge and insights necessary to embark on a successful cloud migration journey.

# Table of Contents

1. [Migration Strategy](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#5252)
2. [Cloud Migration](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#e678)
3. [Cloud Transformation](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#9b37)
4. [The Origin of R Models](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#6026)
5. [The “7 Rs” in Detail](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#290c)
6. [When to Use Each Migration Model](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#a055)
7. [Real-Life Examples Using R Models](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#6ca7)
8. [Final Thoughts](https://medium.com/@rogernem/understanding-the-7-rs-57db362d74d9#accb)

# Migration Strategy

A migration strategy is a detailed plan or approach that outlines how an organization will transition from one system, application, platform, or environment to another (our focus being on the cloud). It involves the careful planning and execution of the migration process to ensure a smooth and successful transition with minimal disruption to business operations.

Migration strategies commonly address the following aspects:

1. **Assessment and planning**: This involves evaluating the existing system, identifying the components to be migrated, assessing dependencies, and determining the best approach for migration.
2. **Risk assessment**: Identifying potential risks and challenges associated with the migration and developing mitigation strategies.
3. **Data migration**: Planning for the transfer of data from the existing system to the new system, ensuring data integrity, and minimizing downtime.
4. **Application migration**: Addressing the migration of applications, including re-platforming, re-hosting, or re-architecting as necessary.
5. **Testing**: Developing a comprehensive testing plan to validate the migrated system’s functionality, performance, and security.
6. **Training and change management**: Planning for the training of staff and stakeholders on the new system, as well as managing the organizational change associated with the migration.
7. **Rollback plan**: Developing a contingency plan to revert to the original system in case of migration failure.

Overall, a migration strategy aims to ensure that the migration process is well-managed, that potential risks are mitigated, and that the new system or environment meets the organization’s requirements and expectations.

# Cloud Migration

It is the process of moving digital assets of an organization, such as data, applications, and IT infrastructure, from on-premises or traditional computing environments to \*cloud-based platforms.

Companies can move their whole or partial on-premises set-up to the desired public cloud service providers or build a private or hybrid cloud computing setting.

The primary goal of cloud migration is to shift existing workloads to the cloud to take advantage of its scalability, and flexibility, and potentially reduce operational costs.

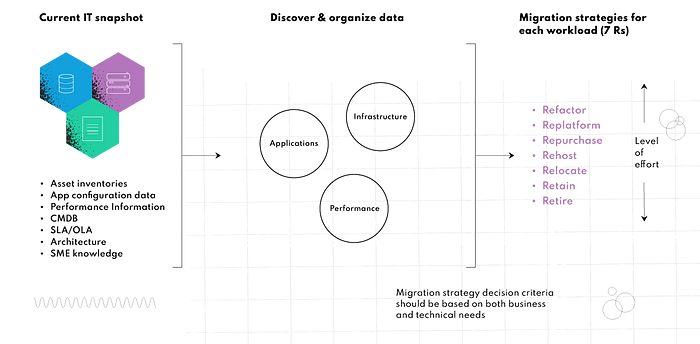
In cloud migration, the existing systems are replicated without making any significant changes to their architecture or functionalities.

\* Remote servers and services accessed over the internet, typically provided by third-party cloud providers like Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and others.

# Cloud Transformation

It is a more comprehensive approach that involves not just moving existing systems to the cloud, but also seeks to modernize applications by **restructuring, optimizing, and sometimes even rewriting** them to take full advantage of cloud services, architecture, and practice and fully leverage cloud-native capabilities.

This can include rearchitecting applications to be more modular, scalable, and fault-tolerant, adopting microservices, using serverless computing, and embracing DevOps practices.



— High-level Workflow of Cloud Migration Strategy

# The Origin of the R Models

## Gartner’s 5 Rs Model

The concept of using R Models in cloud migration strategies originated from Gartner in 2010 when they introduced the \*[5 Rs migration strategy](https://www.gartner.com/en/documents/1485116), which provided a framework for organizations to classify their applications based on their suitability for migration to the cloud.

**Gartner 5 Rs migration strategy:** Rehost (lift-and-shift), Refactor (re-architect), Revise (re-platform), Rebuild (re-develop), Replace (drop and shop)

## AWS 6 Rs Model

Recognizing the need for a more comprehensive migration framework, Amazon Web Services (AWS) expanded on Gartner’s 5 Rs model by adding the sixth R, **Retire**. This additional strategy highlights the importance of evaluating the ongoing value of existing applications and identifying opportunities to decommission or replace legacy systems with more efficient cloud-based alternatives.

**AWS 6 Rs migration strategy:** Rehost (lift-and-shift), Replatform (lift-tinker-and-shift), Repurchase, Refactor (re-architect), **Retire**.

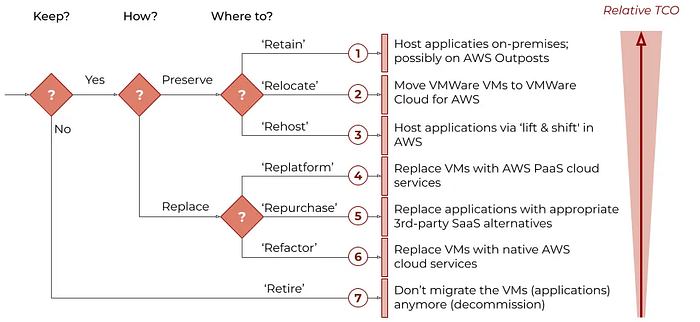
## AWS 7 Rs Model

As cloud computing has continued to mature, and organizations have become more sophisticated in their approach to migration, AWS introduced the 7 Rs model. This latest iteration of the migration framework adds the **Retain** strategy, which recognizes that not all applications and data can or should be moved to the cloud.

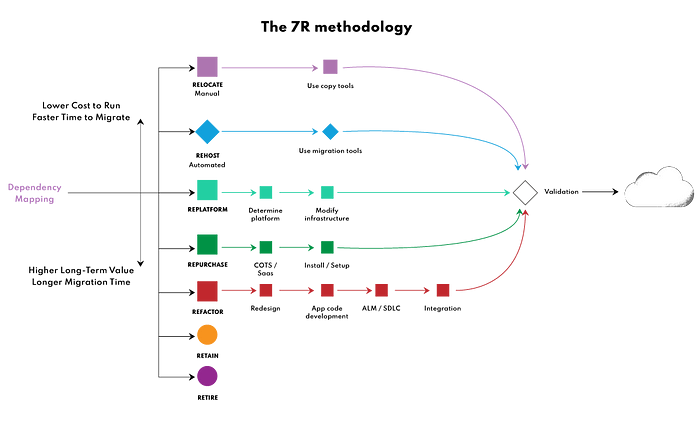
By including the option to retain workloads on-premises, the 7 Rs model provides organizations with even greater flexibility in their migration planning. This allows businesses to balance the benefits of cloud computing with the need to maintain control over critical systems and comply with regulatory requirements.

**AWS 7 Rs migration strategy:** Rehost (lift-and-shift), Replatform (lift-tinker-and-shift), Repurchase, Refactor (re-architect), Retire, **Retain**.

# The “7 Rs” in Detail



— The 7 Rs strategies for migrating workloads to AWS



— The 7Rs migration strategy (based on AWS documentation)

Let’s look into the “7 Rs” cloud migration strategy, according to the expanded model introduced by AWS, based on the original “5 Rs” model introduced by Gartner:

## Retire (Stop using)

This strategy is used when terminating or downsizing legacy applications that are no longer supported by the vendor or useful in production and/or the hassle and expense of migrating to the cloud may not be worthwhile at all. In such instances, business-critical workloads that operate on inefficient legacy frameworks are safely retired as the first step towards the adoption of modern, cloud-native deployments.

## Retain (Do nothing now and revisit it later)

This strategy is applicable to applications facing challenges in migrating to the cloud due to incompatibility for example, legal or privacy restrictions, prohibitive costs associated with rehosting, replatforming, refactoring, rebuilding, or relocating, latency concerns, and various other reasons. In such cases, these applications may not be suitable for retirement and should persist in their existing framework.

Enterprises typically decide to retain a workload if it relies on another application that needs to be migrated first or when there’s no immediate business value in migrating the application to the cloud. As for vendor-based applications, an enterprise may also choose to retain if the service provider plans to eventually release a Software-as-a-Service (SaaS) model.

## Rehost (Lift and Shift)

The rehost migration strategy is often the simplest way to migrate and it involves leveraging cloud Infrastructure-as-a-Service (IaaS) offerings to redeploy workloads on a cloud instance. The strategy allows enterprises to move an on-prem application and all its dependencies as-is to the cloud.

Without changing the core infrastructure, this approach allows organizations to transfer all application data and workflows to cloud services that match the workload’s existing storage, networking, and compute requirements. Since operational and configuration constructs of workloads remain intact, the rehost strategy is also easy to perform and is suitable for enterprises that lack in-house cloud-native expertise.

## Relocate (Hypervisor-Level Lift and Shift)

This strategy involves migrating workloads without impacting ongoing operations, rewriting the application source code, or acquiring new hardware. With this strategy, an enterprise can migrate a collection of servers from an on-premises platform, such as Kubernetes or VMware, to a cloud version of the same platform (e.g. managed Kubernetes services like GKE — Google Kubernetes Engine and EKS — Amazon Elastic Kubernetes Service).

Relocating minimizes downtime and disruption since clients remain seamlessly connected during the migration process. As this strategy doesn’t require significant changes in the configuration and architecture of workloads, it’s not necessary to retrain staff or invest in upgraded hardware, thereby reducing operating expenses. It also makes migration costs more predictable by placing clear limits on scalability.

## Repurchase (Drop and Shop)

This strategy involves swapping internally administered systems for third-party managed services commercially available from cloud providers, helping teams to retire legacy systems and to move to a consumption-based, Software-as-a-Service (SaaS) subscription model that ties IT costs to generated revenue.

As the services are built and managed by third-party vendors, the repurchase model reduces operational efforts toward managing infrastructure for in-house teams.

The repurchase option also simplifies and expedites migration while reducing downtime and enhancing scalability and efficient regulatory governance. As the migration approach fully leverages cloud-native capabilities, it’s mostly leveraged for workloads that require enhanced application performance and user experience while minimizing operational overheads.

## Replatform (Lift, tinker and shift or lift and reshape)

While this migration strategy may entail a significant investment of time and resources, it is frequently justified by the benefits it offers. Notably, this strategy enhances the flexibility, agility, and resilience of workloads, empowering them with cloud-native capabilities. Enterprises can smoothly migrate applications to the cloud, integrating platform optimizations to leverage these capabilities.

This approach affords organizations the flexibility to select components for modernization in line with the latest technological trends, all while preserving the application’s source code and core architecture. This ensures the continued operation of legacy applications, bolstering cloud-based compliance and security measures. Consequently, it elevates application agility and optimizes return on investment (ROI).

Since the application’s architecture and functionality are retained, teams don’t require extensive training to operate the migrating workloads.

## Refactor (Re-architect)

Widely acknowledged as the most intricate migration choice, refactoring entails the comprehensive re-architecting of workloads to seamlessly incorporate cloud-native capabilities. Despite demanding a substantial investment in effort and resources, this strategy stands out as the most forward-looking migration approach. By adopting this approach, applications gain the ability to support advanced features, including serverless computing, autoscaling, and distributed load balancing, positioning them to thrive in evolving technological landscapes.

Refactoring helps break down a monolithic application into microservices to achieve high availability and enhanced levels of automation that are often complex to achieve with in-house deployments. While rearchitecting applications for service-oriented architecture may turn out to be costly during the migration phase, a well-planned resulting framework’s operating costs are substantially lower than operating the legacy framework.

# When to Use Each Migration Model

Below is a comparison of the most appropriate use-case and some pros and cons for each migration model:

## ****Retire****

**Use-case:** Redundant workloads and legacy applications no longer in use.  
**Pros:   
-** Requires the least investment in cost, time, and effort  
- Eliminates IT spend on idle resources  
**Cons:**Premature or unplanned retiring of workloads may result in incompatibility with interconnected stacks

## ****Retain****

**Use-case:** Best for organizations looking to exercise control over their resources and those considering a hybrid cloud migration. Also suitable for applications that are required to run on local data centers for compliance or security.  
**Pros:   
-** Saves cloud resources by retaining inefficiencies on-premises.  
- Allows for evaluation of (recently upgraded) applications in order to identify those that require immediate migration or those that could be migrated at a later date.  
**Cons:  
-** Lack of cloud-native capabilities.  
- Requires costly and continuous maintenance.

## ****Rehost****

**Use-case:** For companies looking to expedite cloud migration at a fraction of the cost, while looking to eventually add further changes.  
**Pros:  
-** Requires minimal risk and disruption.  
- Minimizes issues concerning compatibility, disruption, long migrations, or long-distance data replications.  
- Increase reliability, resilience and scale applications without costly upgrades.  
- Easier to integrate and to optimize or re-factor applications once they have been rehosted to the cloud.  
- Enables complete transfer of legacy workloads.  
**Cons:  
-** Introduces operational and technical incompatibilities, affecting user experience.  
- Limited number of cloud-native capabilities.

## ****Relocate****

**Use-case:** Applications running on VMware and local Kubernetes distributions.  
**Pros:  
-** Quickest migration strategy.  
- Requires minimal staff training.  
- Reduces data center operational costs.  
- Requires no change in operational processes for migrated workloads.  
**Cons:  
-** PaaS services can be expensive.  
- Scaling down instances to save on costs becomes a very challenging task.

## ****Repurchase****

**Use-case:** Best for organizations looking to leverage cloud-native capabilities without having to design systems from scratch.  
**Pros:  
-** Quick adoption of cloud-native capabilities.  
- Reduces existing applications’ licensing, infrastructure, and maintenance issues and costs.  
**Cons:**  
- Updates and releases are performed at the vendor’s cadence.  
- Costs can accumulate quickly for repurchasing multiple applications.  
- The need to train users and teams on new systems.  
- New products might not fully resemble current processes.  
- Vendor lock-in.

## ****Replatform****

**Use-case:** For organizations considering a move to the cloud but are concerned about the risks involved in comprehensive migration of legacy apps in one go.  
**Pros:  
-** Offers the ability to choose specific features for maximum ROI.  
- No extensive training required.  
- Enables IT teams to proactively monitor the effectiveness of cloud-native capabilities before migrating additional workloads.  
**Cons:  
-** Changes are costly and can be very time-consuming.  
- May lead to reduced application availability during the migration phase.

## ****Refactor****

**Use-case:** Best for complex applications with high-usability and a strong business case for performance optimization or can no longer address business needs or customer demands due to its limitations or cost.  
**Pros:  
-** Enables organizations to achieve end-to-end cloud-native capabilities of the well-architected framework.  
- Ensures business continuity.  
- Offers personalized levels of automation, scaling, and high availability.  
**Cons:  
-** Relies on extensive staff training and cloud expertise.  
- Requires thorough planning, budgeting, and execution.  
- More complex to manage, requiring continuous monitoring for cost optimization.

# ****Real-Life Examples Using R Models****

The following are some real-life examples of successful Cloud Migrations using the different R Models where organizations have reaped significant benefits while addressing challenges along the way:

* **Netflix (Streaming):** They embraced the rehost (lift-and-shift) approach to migrate its massive infrastructure from on-premises data centers to Amazon Web Services (AWS) which allowed them to scale quickly, improve reliability, and reduce costs.
* **Capital One (Banking):** They opted for the repurchase model, replacing its legacy systems with AWS marketplace solutions, leading to improved agility and faster time-to-market for new products and services.
* **Airbnb (Rental Platform):** They utilized the rebuild (re-develop) approach to modernize its infrastructure and embrace a microservices architecture, allowing them to improve scalability, enhance performance, and introduce new features quickly.
* **Expedia (Travel):** They adopted a hybrid approach, leveraging the refactor (re-architect) and repurchase strategies, allowing them to optimize its applications for AWS while also taking advantage of pre-built solutions.

# Final Thoughts

The right choice between Relocating, Rehosting, Replatforming, Repurchasing, Refactoring, Retaining, or Retiring an application will depend on its size and complexity, hardware requirements, scalability and availability requirements, technical compatibility with the cloud infrastructure, and the company’s budget, among other factors.

In any case, a well thought-out migration strategy will play a key role in successful cloud migration, reducing the risk of issues and **cloud migration challenges** along the way and increasing ROI.

Cloud migration is a big project that requires careful planning and consideration but many companies have already paved the way through trial and error and produced some helpful pathways codified in the evolving “R Models” and if a cloud migration project is in your company’s future, leveraging them is a very smart move.

**If you enjoyed this article and found it helpful, please don’t forget to leave a heart** ❤**, comment** 💬**, clap** 👏🏻**, and share** ➦ **it to show your support.**

**Also, don’t forget to** [**follow me**](https://medium.com/@rogernem) **for more articles. Thank you!**

References:  
- <https://www.gartner.com/en/documents/1485116>  
- <https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/>  
- <https://docs.aws.amazon.com/prescriptive-guidance/latest/application-portfolio-assessment-guide/iterating-7-rs-migration-strategy-selection.html>  
- <https://docs.aws.amazon.com/prescriptive-guidance/latest/application-portfolio-assessment-guide/prioritization-and-migration-strategy.html>  
- <https://docs.aws.amazon.com/prescriptive-guidance/latest/large-migration-guide/migration-strategies.html>