New Relic.

White Paper

Modernizing with AWS:

Getting Started with 'Observability-Driven' Rehosting

Part 1 of our 3-part series on legacy app modernization in the cloud

Introduction

To gain agility and competitive advantage, modernization is a business imperative that no company can afford to ignore. This includes modernizing your applications, infrastructure, technology, and even your approach to software development.

From an infrastructure perspective, modernizing from an on-premises or legacy environment to Amazon Web Services (AWS) helps you begin to harness the agility, flexibility, and scalability of the cloud. Modern software development methodologies such as DevOps go hand-in-hand with modern cloud environments, making it possible to take advantage of cloud technologies and tools to further accelerate software development and optimize the elasticity, resiliency, and performance of software running in the cloud.

However, legacy applications must be modernized as well, or they will hinder your ability to use the cloud to achieve transformative outcomes for IT and the business. Of the six approaches to application modernization (retire, repurchase, retain, rehost, replatform, and refactor), rehosting is the most common first step. That's because rehosting offers a relatively low-risk way to begin modernizing and reaping early benefits such as lower costs, improved performance, and the ease of operation on a modern platform.

If the thought of rehosting applications seems daunting, that's understandable. While the risks involved in rehosting are relatively low depending on the criticality of the application, there are nonetheless risks that you should work to minimize. Mitigating those risks requires proper planning, observability, data-driven decision making, and the adoption of best practices such as those in the AWS Well-Architected Framework, which was developed to help companies build secure, high-performing, resilient, and efficient infrastructure for their applications.

This white paper introduces the "why" and "how" of rehosting applications on AWS and sheds light on how to apply the five pillars of the AWS Well-Architected Framework so your rehosting effort optimizes your benefits.

> "... currently, 65% of North American enterprises rely on public cloud platforms."
>
> Source: "Predictions 2020: Cloud Computing Sees New Alliances And New Security Concerns," Forrester, November 2019

What Is Rehosting and When Should You Use It?

Let’s start by explaining what rehosting means. It’s a relatively low-risk approach to modernizing an application that involves “lifting” the application as is, then “shifting” it over to a new environment—without making changes to the business logic, and with only minimal or no code changes to enable the application to run in the new environment.

Rehosting can be as simple as moving an application to a newer version of the same infrastructure or it can be more complex, such as moving from an on-premises server to AWS. Examples of rehosting include:

- Moving an application from an older Amazon Elastic Compute Cloud (EC2) instance to a newer one
- Migrating an application from on-premises infrastructure to AWS
- Moving/migrating an application from a different cloud environment to AWS

Making the decision about how to modernize an application requires a deep understanding of the application, the infrastructure it currently runs on, its quality and performance, and the application’s impact on the business. Armed with an in-depth view of the application, you can then decide whether rehosting or some other modernization approach is the best choice.

While there are many compelling reasons for choosing to rehost an application, most often organizations decide to rehost based on common factors and characteristics such as those listed in Table 1.

When the application is ...	Then rehosting is the right choice because ...
Complex and/or a critical system	It’s the fastest approach to modernization with the least amount of risk (besides doing nothing).
Part of a group of applications being rehosted	The application has dependencies on other applications being rehosted.
A good candidate for automation	Cloud automation can reduce or eliminate time-consuming, manual efforts needed to operate the application.
One that experiences large changes in volume (such as seasonal spikes)	The application can benefit from compute elasticity in the cloud.
No longer able to run in the current environment	The application must move due to hardware being retired, the data center being decommissioned, etc.
A black-box application	There is no way to change the business logic as with other modernization approaches.
A static application	No code changes have happened or are planned for the application, but it can still benefit from moving to a new environment.

Table 1. Application Characteristics and Why Rehosting Is the Right Approach to Modernizing Certain Applications

Know the Risks and How to Minimize Them

Even though rehosting is a relatively low-risk approach to modernization, as with any IT project, there are always some risks involved, and it’s critical to anticipate and minimize them. Before you begin, it’s important to understand all the possible risks of moving an existing application, which could include:

- Choosing the wrong application to rehost, which prevents you from meeting your project goals
- Losing visibility into application health and performance
- Having part, or all, of the application break in the new environment
- Increasing cloud spend beyond expectations because the application inefficiently consumes cloud resources
- Facing unplanned costs from fixing application issues that happen during the rehosting effort
- Not achieving measurable benefits after the rehosting
- Migrating existing problems with the application to the new environment

The good news is that all of these risks and many others can be mitigated with the right amount of planning, observability, and in-depth data derived before, during, and after the rehosting effort.

Take an Observability-Driven Approach

While successfully rehosting one or more applications requires forethought and planning to mitigate risk and achieve your organization’s business and IT goals, it also takes deep insight into the state of the application.

Observability gives you comprehensive baseline data so you can make informed decisions about rehosting an application that reduce your risk, streamline the effort, and give you the means for measuring success.

One of the critical aspects of an observability-based approach to rehosting is how it impacts the end user. To increase the likelihood of achieving a positive impact (such as improved performance and reliability), you’ll need data that helps you understand:

- The current, “normal” end-user experience
- Ongoing application issues to be prioritized and resolved before rehosting
- Application dependencies
- Brittleness of the application
- Target goals for customer experience

Next, begin gathering data in the context of the AWS Well-Architected Framework, as shown in Table 2. This information helps guide you in prioritizing applications for rehosting and setting goals your organization wants to achieve with the rehosting effort.

Pillar	Questions and Metrics
Cost optimization	• Where are you spending the most? • What is your infrastructure cost per application user?
Reliability	• What is the rate of application uptime? • How often are your users impacted by errors? • How well are scaling events working?
Performance efficiency	• How efficient is resource usage? • Do you need better regional alignment with users? • How quickly can you deploy globally?
Operational excellence	• How long does it take you to provision environments? • How frequently do you provision? • How long does it take to resolve problems?
Security indicators	• Are operating systems and/or patch levels causing security concerns?

Table 2. Using the Pillars of AWS Well-Architected Framework to Prioritize Rehosting Efforts

Using the information you've collected, you can then formulate your rehosting plan by designating:

- Which applications will be rehosted and in what order
- Which issues will be addressed ahead of the rehosting
- What the plans are for testing and user acceptance
- What your technical and business goals are for each application, for example:
    - Better end-user experience (improved performance and reliability)
    - Lower operating costs or the ability to decommission hardware/ data center
    - Faster time-to-market with new features
    - Improved scalability

To learn more about how to collect application baseline data, review these tutorials:

- Guide to Cloud Adoption: Create Application Baselines
- Optimize Your Cloud Native Environment: Establish Objectives and Baselines

Use Observability Before, During, and After Rehosting

Observability is not only imperative for the planning of your rehosting exercise and decision-making ahead of the initiative, but it’s also the best way to gain the confidence and insight to successfully move your application to its new environment. By maintaining complete visibility into the application, infrastructure, and end-user experience throughout the rehosting effort, you can accelerate your efforts without increasing risk, anticipate and avoid problems, minimize complexity, and maximize the benefits of your cloud environment.

Here’s how to use observability at each phase of the rehosting effort:

- **BEFORE:** Once you’ve selected the applications to rehost, begin instrumenting them for observability in their current environment using New Relic One to understand application performance, end-user experience, resource consumption, error rates, application uptime, and other critical KPIs. These measurements will serve as baseline metrics for comparison during and after the rehosting. This data also allows you to create a plan for addressing systemic application issues that you’ve identified.
- **DURING:** As you begin the rehosting process, it’s important to test your applications to make sure the transition isn’t causing any problems or slowdowns. During the rehosting, use New Relic One to check the performance of application code, databases, and external services and compare it against the pre-rehosting baselines you established before. This gives you an apples-to-apples comparison.
- **AFTER:** Once the rehosted applications are running in the new environment, use New Relic One to measure your modernization success and identify opportunities to further optimize the applications for performance gains, efficient use of resources, and ease of operation. Comparing before and after baseline data also helps you minimize the amount of time and money spent supporting dual infrastructures while you confirm that everything is working as it should in the new environment.

Observability Accelerates Cloud Deployments

A Total Economic Impact™ study conducted by Forrester Consulting in 2019 finds that customers leveraging the New Relic platform deploy applications to the cloud 90% faster and reduce more than 90% of the cost associated with the deployment of applications in cloud environments.

Read the study

Pillar: Cost Optimization

Using New Relic One helps you stay on track and on budget for your rehosting effort. One example is by reducing the amount of time you need to run dual infrastructures. New Relic One also helps by giving you a clear understanding of existing application dependencies, issues, and performance before you begin rehosting, so that you spend less time during the rehosting effort dealing with unsurfaced application issues.

New Relic can also help you make better decisions about sizing your cloud instances to avoid wasting budget spend on cloud resources that are underutilized by the application. For instance, using data in New Relic One, you can configure your autoscaling policies properly to manage instance sizes to prevent overprovisioning and unnecessary spend.

Pillar: Reliability

A reliable system is one that is stable, predictable, and highly available. From an organizational and infrastructure perspective, it’s the ability to prevent failures and quickly recover from any that do occur, with the goal of delivering great customer experiences.

To understand how rehosting impacts the reliability of an application, compare your baseline data to the actual information you gather during and after the rehosting, including:

- Apdex score (user satisfaction)
- Mean time to resolution (MTTR)
- Errors and where they are surfacing
- Reworked tickets
- Any infrastructure-as-code with issues filed

Optimize Computing Costs in the Cloud

“Traditional enterprise data centers are equipped with finite, pre-procured, capital expenditure (Capex)-oriented capacity but a more efficient use of that capacity does not automatically translate into cost savings. The cloud computing model reverses this paradigm.”

Source: “How to Manage and Optimize Costs of Public Cloud IaaS and PaaS,” Gartner, March 2020

Pillar: Performance Efficiency

Performance efficiency is about using computing resources efficiently to meet system requirements and maintain that efficiency as demand changes and technologies evolve. To optimize how rehosted applications use resources, start by focusing on your baseline performance data and compare it against how the application is performing in the new AWS environment. Specifically, you can compare the Apdex score, page load times, slow transactions, and error rates as shown in Figure 1.

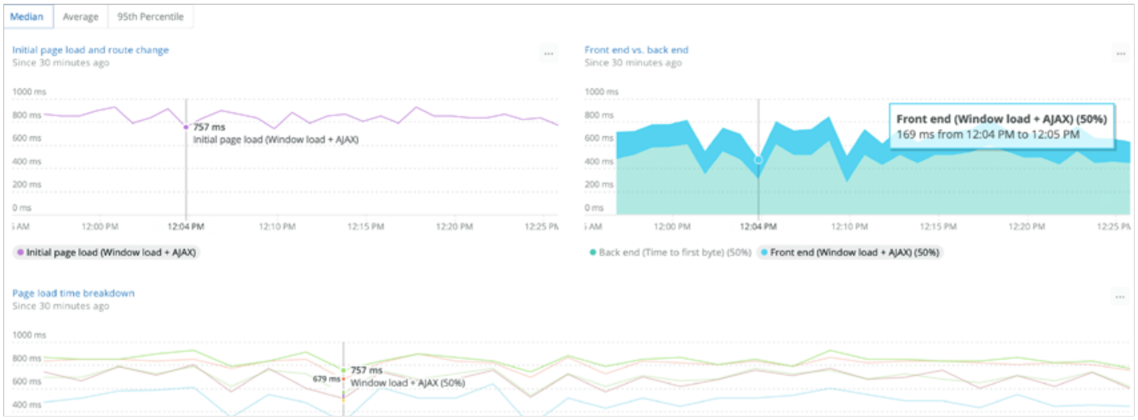


Figure 1. Dashboard Showing User Experience Metrics

Another consideration is locality to users: Are you aligned regionally with where your users are located and, if so, what kind of performance are they experiencing in each location? AWS gives you the ability to expand globally in minutes, to locations around the world, to take advantage of different cost structures and geographies for efficiently serving your customers, as shown in Figure 2. You can choose to locate resources based on user location and business goals.

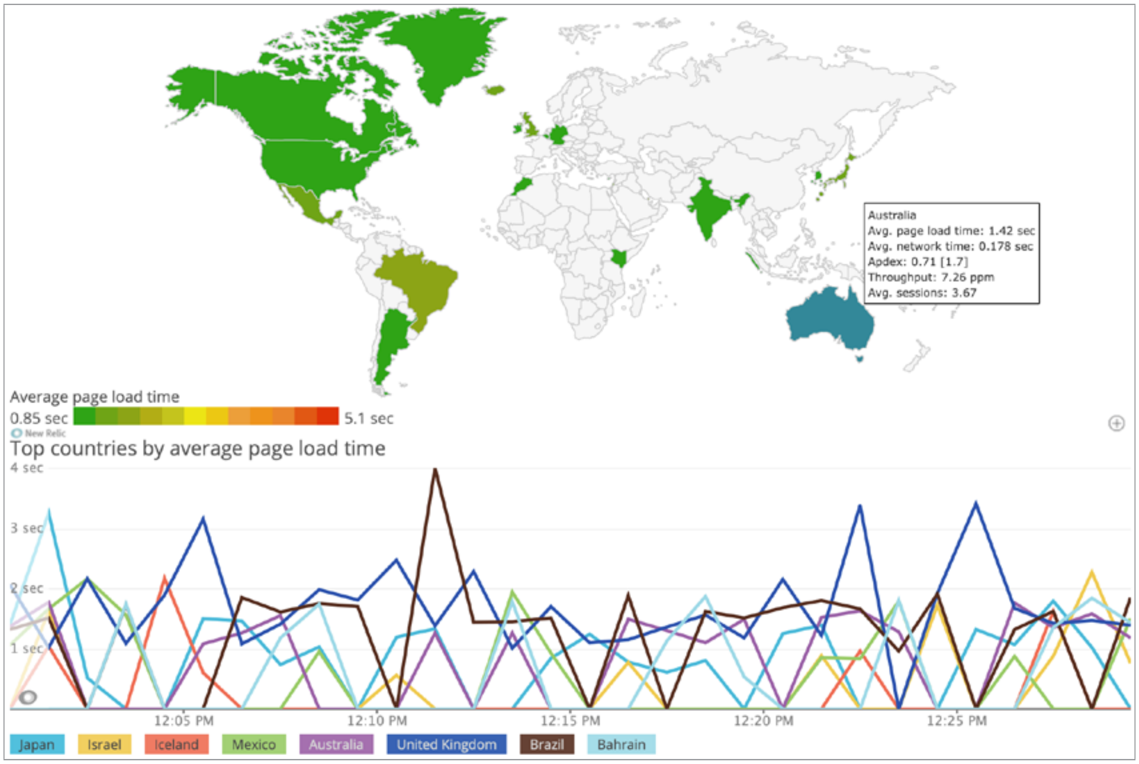


Figure 2. Determining Locality to Users: Average Page Load Time by Country

You should also review resource consumption, such as CPU and memory usage, in the application's new environment. Storage is an important consideration as well. Beyond simply understanding capacity, consider read and write patterns to choose the best storage option for your application's usage profile. This information will help you rightsize instances to optimize performance efficiency in AWS.

Pillar: Operational Excellence

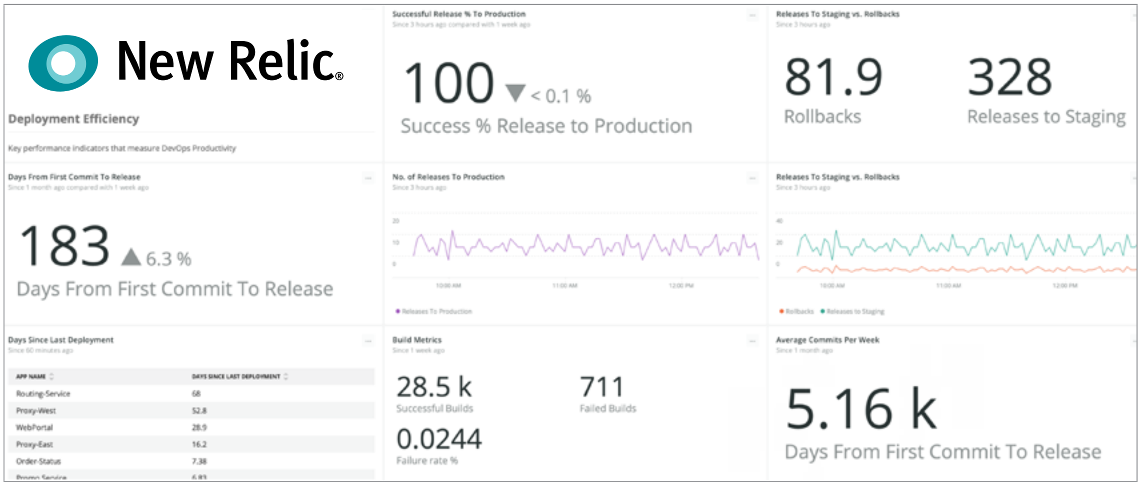
Embarking on a rehosting effort is also a great time to start or continue adopting strong practices that support maximizing your cloud investment. According to DevOps research firm DORA, the use of the cloud improves software delivery performance and availability, and teams that leverage cloud computing's essential characteristics are 24 times more likely to be high performers.[1]

No matter where you are on your DevOps journey, focus on improving your processes around key DevOps capabilities such as automation, continuous integration, continuous delivery, continuous testing, continuous monitoring, and rapid remediation. Monitoring areas such as AWS

[1] "Accelerate: State of DevOps 2019," DORA, 2019

CloudFormation and Amazon Virtual Private Cloud (VPC) events as well as tracking how quickly changes are happening in your software will give you indications of how brittle your new environment is and how quickly your team is adopting new processes required for success.

With data-driven DevOps powered by New Relic One, you get observability, visibility, and detailed metrics to help you maintain performance and availability even as the pace of change and releases increases. New Relic One's deployment tracking, notes, plugins, and other capabilities help consolidate telemetry data into a single dashboard for tracking your process improvements along with your rehosting efforts, as shown in Figure 3.


New Relic

Deployment Efficiency

Key performance indicators that measure DevOps Productivity

Successful Release % To Production
Since 3 hours ago compared with 1 week ago

100 ▼ < 0.1 %
Success % Release to Production

Releases To Staging vs. Rollbacks
Since 3 hours ago

81.9
Rollbacks

328
Releases to Staging

Days From First Commit To Release
Since 1 month ago compared with 1 week ago

183 ▲ 6.3 %
Days From First Commit To Release

No. of Releases To Production
Since 3 hours ago

Releases To Staging vs. Rollbacks
Since 3 hours ago

Days Since Last Deployment
Since 60 minutes ago

APP NAME	DAYS SINCE LAST DEPLOYMENT
Routing-Service	68
Proxy-West	52.8
WebPortal	28.9
Proxy-East	16.2
Order-Status	7.38
Promo-Service	6.83

Build Metrics
Since 1 week ago

28.5 k
Successful Builds

711
Failed Builds

0.0244
Failure rate %

Average Commits Per Week
Since 1 month ago

5.16 k
Days From First Commit To Release

Figure 3. DevOps Success Metrics

Pillar: Security Indicators

Security processes and best practices should be incorporated throughout the software development life cycle to protect information and systems from attack. This life cycle includes rehosting, in which SecDevOps plays an important role in automating security into the continuous integration and continuous delivery model for modernized applications.

Security indicators important for rehosting include the operating systems and their versions, as well as the Amazon Machine Images (AMIs). Security architects will want to know whether an underlying system is out of date and whether there are security alerts filed against it that impact the application.

Data points to consider as you prioritize applications to rehost and prepare to select your new environment include:

- Out-of-date operating systems
- Too many different operating systems and versions being supported
- Instances in the default Amazon Virtual Private Cloud
- AMIs used to create the instances

To learn more about DevOps success and how to measure it, read this tutorial: Guide to Measuring DevOps Success.

Armed with this information, you can get guidance from your security team about which hosts should be prioritized for rehosting based on any security issue. To easily find the hosts that need to be updated, use the filter ability within New Relic One. If your hosts are in the AWS cloud, you can filter by the additional tags available with the New Relic integration. These include instance type, region, or custom tags such as application or state.

Next Steps

Once you’ve completed an application, it’s time to evaluate whether the rehosted application is also a candidate for further modernization. For instance, replatforming (taking a component of an application such as a database and moving it to a managed service) could deliver additional time and resource savings. Refactoring (making code-level changes to take advantage of modern cloud services, architectures, and technologies) could improve the quality, performance, scalability, reliability, and flexibility of the application and enable the rapid delivery of new features.

By using New Relic One—the most powerful cloud-based observability platform—throughout your continuous modernization and optimization efforts, you can modernize your applications faster, more cost-effectively, and with less risk and begin to reap the benefits of moving to the cloud. As an AWS Advanced Technology Partner, New Relic has been granted AWS Competency in six solution areas (Migration, DevOps, Containers, Mobile, Retail, and Government) based on our technical proficiency and proven customer success.

To learn how New Relic One can help you in rehosting or any phase of application and infrastructure modernization, visit newrelic.com/aws.