



Measure Twice, Cut Once

Cloud Migration Measurement And Acceptance Testing With New Relic



Table of Contents

| | |
|-------------------------------------|-----------|
| INTRODUCTION | 03 |
| CLOUD MIGRATION ACCEPTANCE CRITERIA | 03 |

| | |
|---|-----------|
| THE MEASUREMENT TOOLBOX: NEW RELIC DIGITAL INTELLIGENCE PLATFORM | 04 |
|---|-----------|

| | |
|--|-----------|
| STARTING WITH THE BASICS: BLACK-BOX BASELINES | 05 |
|--|-----------|

| | |
|--|-----------|
| ADVANCED MIGRATION DETAILS WITH NEW RELIC | 07 |
| FRONTEND APPLICATION PERFORMANCE | 08 |
| BACKEND APPLICATION PERFORMANCE | 09 |
| INFRASTRUCTURE PERFORMANCE | 10 |

| | |
|--|-----------|
| MY BASELINES ARE ESTABLISHED, NOW WHAT? | 11 |
|--|-----------|

| | |
|---------------------------------------|-----------|
| MOVING FORWARD WITH CONFIDENCE | 13 |
|---------------------------------------|-----------|

Introduction

Now that you've made the decision to migrate, where do you start? What does success mean? How do you ensure you don't regress in performance? There is an old adage in carpentry that you should measure a board twice before you make a cut so you can limit the need to make a second cut correcting a mistake. This same principle applies to migrating your workloads to the cloud; if you're not measuring the state of an application before and after a workload moves then you are likely to end up wasting precious time and effort.

At that moment of truth when the workload arrives on AWS, *seeing* an application running in AWS isn't enough, you need to *know* the application is stable, customer experience is at least consistent, and infrastructure consumption is acceptable. Enter the New Relic Digital Intelligence Platform, a simple, low-impact and powerful set of tools to baseline, compare, and validate your application ecosystem both on-premises as well as in AWS. This simple guide will show you how to get quantifiable measures of success for your migration, ensure that your migration goes smoother when things do go wrong, and enable ongoing quality in your applications long after they've been successfully migrated to AWS.

Specifically, you'll learn:

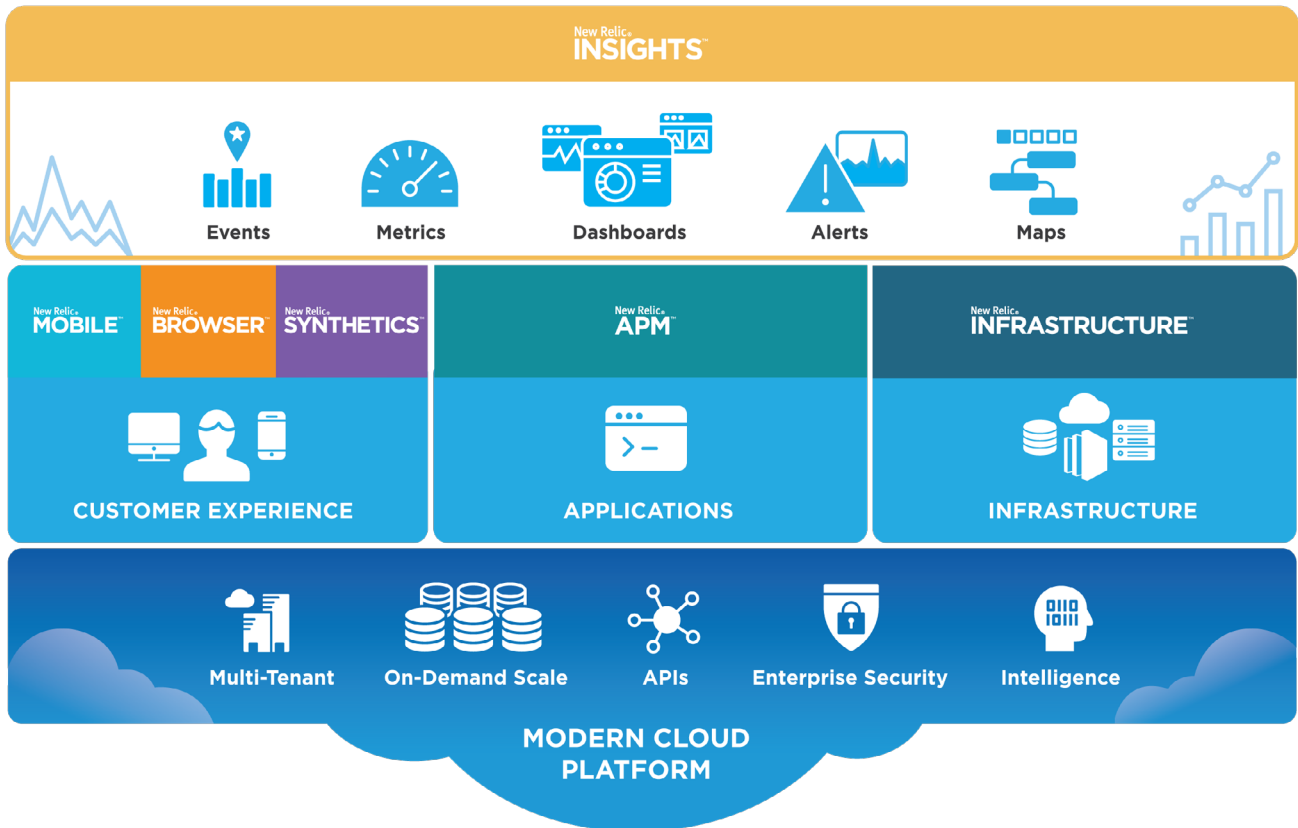
- **Acceptance criteria:** The concept and requirements.
- **The tools:** What New Relic is and how to quickly get it set up on your systems.
- **Basic baselines and acceptance criteria:** How to get immediate "black-box" baselines of your systems to answer the most important questions about your migration in minutes.
- **Granular baselines and acceptance criteria:** How to leverage the full suite of New Relic's Digital Intelligence Platform to get granular insights about how your applications perform at every tier and to understand all the requisite dependencies that need to be considered as part of your migration.
- **Post-migration optimization:** How to continue to benefit from New Relic's Digital Intelligence Platform after your successful migration, guaranteeing that your environments are never again susceptible to performance degradation without you being aware.

Cloud Migration Acceptance Criteria

Employing a measurement and monitoring strategy to establish baseline metrics during the discovery phase ensures validation of workloads post-migration, success post-migration, and the ability to confidently proceed with production applications. Baseline metrics become particularly important if the project involves handoffs between teams such as system integrators who may not participate in every phase. At the highest level, determining if a migrated workload will be stable and healthy on the new platform before proceeding to full production requires teams to be able to show data that answers the following questions:

- Has an application’s performance gotten faster or slower?
- Is an application more or less stable than before?
- Are we losing customers due to either of the previous questions?

The tolerances for what is acceptable will vary by the business criticality and user expectations of an application. This guide will help you prepare and show you what to measure before and after the moment of truth when the application is running on AWS and everyone is wondering: can we proceed?



The Measurement Toolbox: New Relic Digital Intelligence Platform

The New Relic Digital Intelligence Platform is a modular set of tools that grant both deep and broad visibility to your applications and surrounding components running in production. The tools are agnostic to your runtime environment, so whether you’re deploying on-premises or in the cloud, the fundamental setup process is the exact same. This consistency across environments will also allow for an easy apples-to-apples comparison of your systems before and after the migration. Even better, New Relic does not introduce any additional compute dependencies to your system, so you don’t have to worry about installing, maintaining and scaling databases or additional services as part of your migration plan.

The setup process for New Relic is broken down into distinct steps per product which target different tiers of your environment. The following is a high-level overview of the parts of the Digital Intelligence Platform most relevant to AWS migrations with links off to their respective setup documents:

- **New Relic APM** is a development-time dependency that you add to your backend code which will give detailed information about your application such as response time, throughput, and error rate wherever it is running. APM will show you the line of code that an exception was thrown from, the exact database query that's slowing down your system, and even automatically build a map of all upstream and downstream dependencies to your application. Bear in mind that the setup process for New Relic APM will differ depending on the language your particular applications were developed in. [More setup details here.](#)
- **New Relic Browser** is a JavaScript library which is injected into your HTML documents to get detailed information about how your front-end code is executing on your users' browsers. New Relic Browser will show you JS errors broken down by browser, geographic response times and even full session traces of user engagement with your website. New Relic Browser can be setup automatically by simply installing New Relic APM on your backend and turning it on via the web interface or by copy/pasting the JavaScript directly into your HTML pages. [More setup details here.](#)
- **New Relic Synthetics** is a distributed synthetic monitoring tool which will take a URL or Selenium-style script and execute it to validate your systems as if it were a user accessing your content. This tool acts as a 'canary in a coal mine' notifying you of issues long before your users experience them. Synthetics is managed entirely from the New Relic interface and can be setup in a matter of minutes without a single change to your systems. [More setup details here.](#)
- **New Relic Infrastructure** is a daemon agent which runs on your hosts grabbing server level details about the underlying infrastructure supporting your applications such as CPU, memory, and running processes. This tool is equally well-suited for traditional on-premises deploys as well as dynamic, cloud-scale environments. In addition to the daemon agent, there is both a SaaS-to-SaaS integration with AWS to pull in additional data about AWS components as well as an SDK for instrumenting additional on-host systems such as databases, caches, and more. [More setup details here.](#)

Starting With the Basics: Black-Box Baselines

Cloud migrations can take many forms. Some companies choose to simply port their applications directly from their data center to EC2 instances (a "Lift and Shift" migration) while others focus on completely re-architecting the applications to take advantage of benefits only available on the cloud. This spectrum of migrations adds some nuance in the types of instrumentation you'll need at more detailed levels, and many people get lost in the weeds here.

At the end of the day, no matter what your approach, there are generally only three primary questions you want to answer:

- Has my application gotten slower?
- Is my application less stable than before?
- Am I losing customers due to either of the previous questions?

Sure, you want to know your database response times, how long messages are sitting on the queue and much more detailed information about your systems—and New Relic can certainly surface all that data, but the importance of those deep dives are usually dependent on your answer to one of the above questions.

Because of this, we recommend you start by validating your systems as a black box. This is extremely simple when using New Relic Synthetics and New Relic Browser. As these tools are able to test the entire flow of data from an end-user’s perspective including all backend components working in concert. Ultimately this cumulative view of your performance and stability is the primary measure you’re trying to optimize.

Setting up a New Relic Synthetic **Simple Browser** or **Scripted Browser** monitor pointing at your homepage or root experience (see Figure 1) will tell you exactly how long it takes your page or experience to load as well as how often it fails. These ‘checks’ will be run on a regular interval (managed entirely from the New Relic UI) and will give you a historical baseline to compare against. For example, if it takes an average 700ms for your page to load and it has a success rate of 99.9% on-premises, then you now have quantifiable measures to compare against after your migration is complete (see Figure 2).

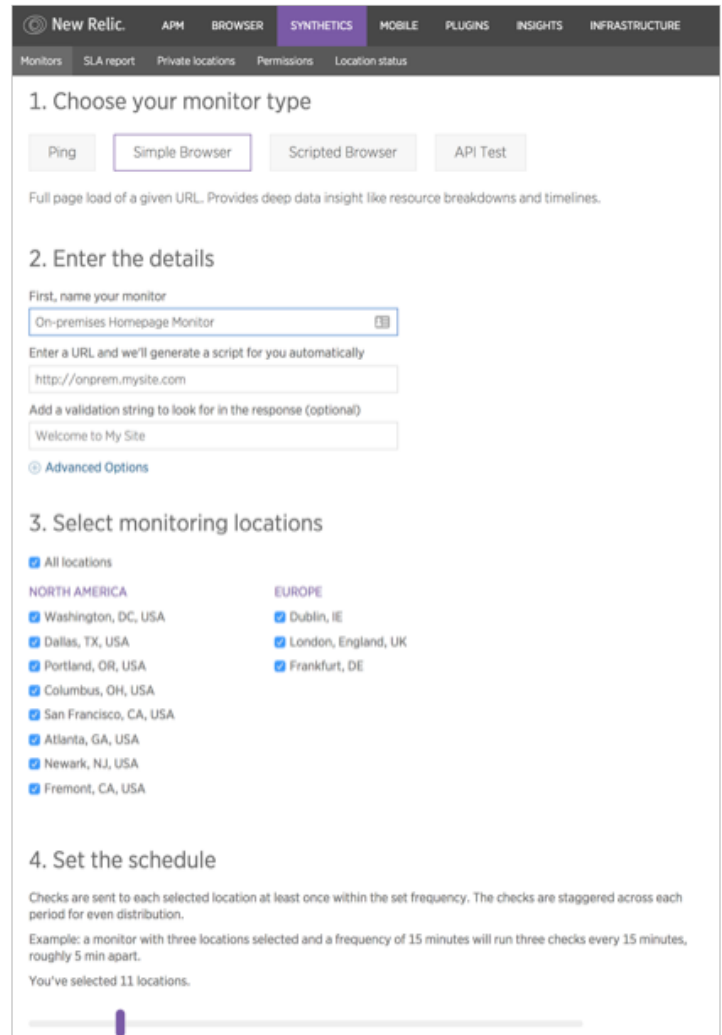


Figure 1: Setting up a New Relic Synthetic monitor takes only a matter of minutes and is managed entirely from the New Relic interface.

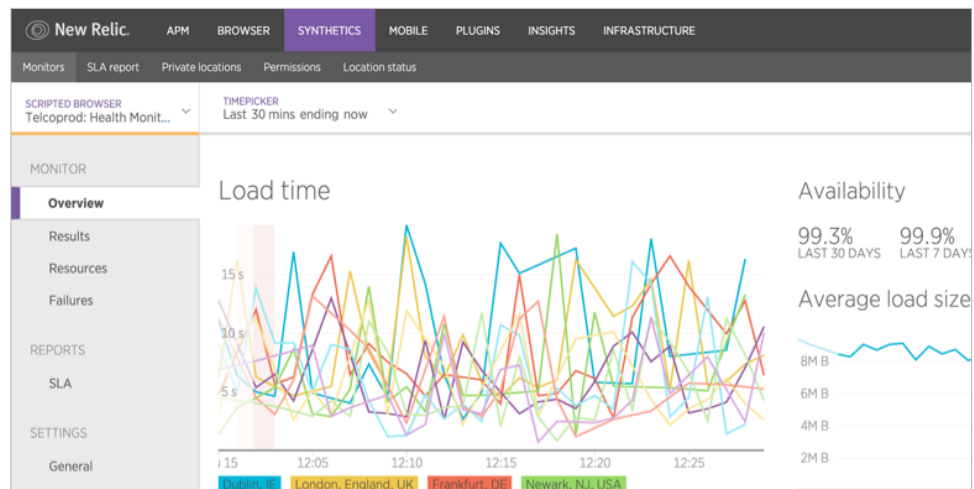


Figure 2: Once a monitor is set up in New Relic Synthetics you'll get a wealth of information and historical data about how your application is performing.

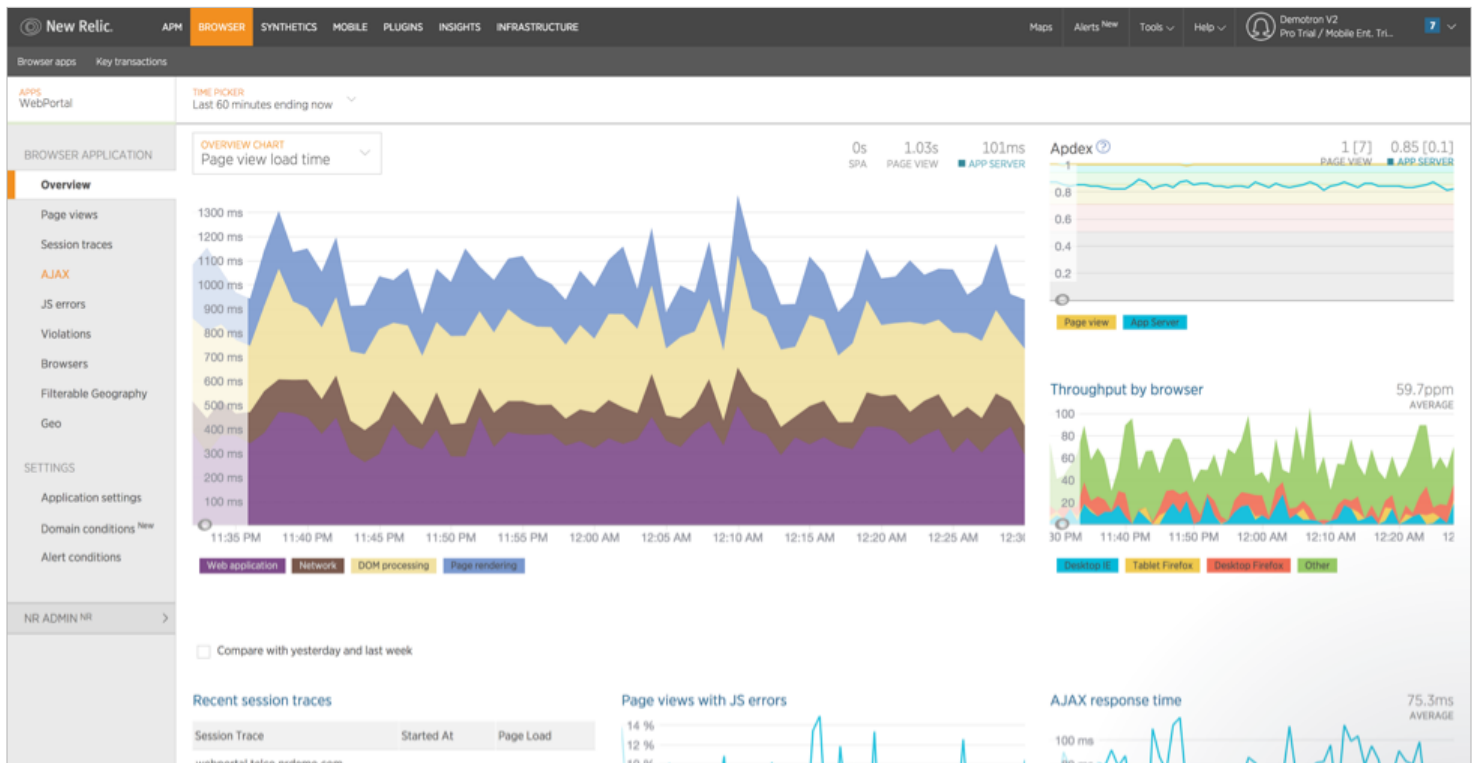


Figure 3: New Relic Browser’s Overview page offers a single-page view to gauge your most critical metrics including response times, throughput and JS errors.

While Synthetics is great for baselining performance from a mock end-user perspective, sometimes you want the raw data of what your customers are actually doing in the wild. Specifically to answer the third question above, “Am I losing customers due to either of the previous questions?”. This is where New Relic Browser shines (see Figure 3). This lightweight JavaScript library will log your application’s throughput and real response times for actual customers (amongst many other things). You can deduce a lot from just the overview page; for example, if you see a spike in response time or error rate and a drop in throughput, then you know immediately that you’re not dealing with a benign problem but are actively losing customers due to your performance issues.

With these two components setup you now know how to ensure that your overall experience doesn’t degrade as part of the migration. Simply setup a duplicate Synthetic monitor for your AWS port and ensure New Relic Browser is turned on and you’re in business.

Advanced Migration Details with New Relic

The previous section focused on how you would use New Relic to get the 10,000ft validation of whether or not your applications have improved or degraded post-migration, but if those numbers skew one way or the other, you often want to know exactly where the slowdowns or instability is occurring.

For the sake of simplicity, we will break this down into three distinct tiers which can be addressed in isolation. We will focus on the most important metrics to track at each tier as well as specific migration scenarios that you are likely to encounter along the way.

Frontend Application Performance

As was previously mentioned, end-user experience is your primary target. Whether that be a browser-based frontend or a mobile app, the story holds true. At the end of the day, each individual component’s response time is only important to you as the maintainer of the code—that is, your customers don’t care about individual components because they only ever see aggregate responses of all the backend components working together. This makes frontend performance baselining extremely important. However, as part of a migration to AWS you are unlikely to re-architect your frontend application. In a typical migration, you can ignore many of the more advanced capabilities of the products focused on optimization and instead focus on only a few core KPIs.

Those key metrics specifically are:

- **Browser:** Page loads, JS Errors, AJAX failures, AJAX response times, geographic response times (see Figure 4)
- **Mobile:** HTTP errors, HTTP response times, app crashes (see Figure 5)

Additionally, as was detailed in the previous section, New Relic Synthetics can provide a more sandboxed view of the same metrics captured by New Relic Browser. The difference being that New Relic Browser represents all of your real customer engagement with your site, but New Relic Synthetics simply emulates a user interaction. One notable place that Synthetics shines in regards to AWS migrations is viewing your static asset load times. Synthetics will show you exactly how long it is taking for each component to load on your website to easily ensure that your CDN (AWS CloudFront) is performing properly (see Figure 6).

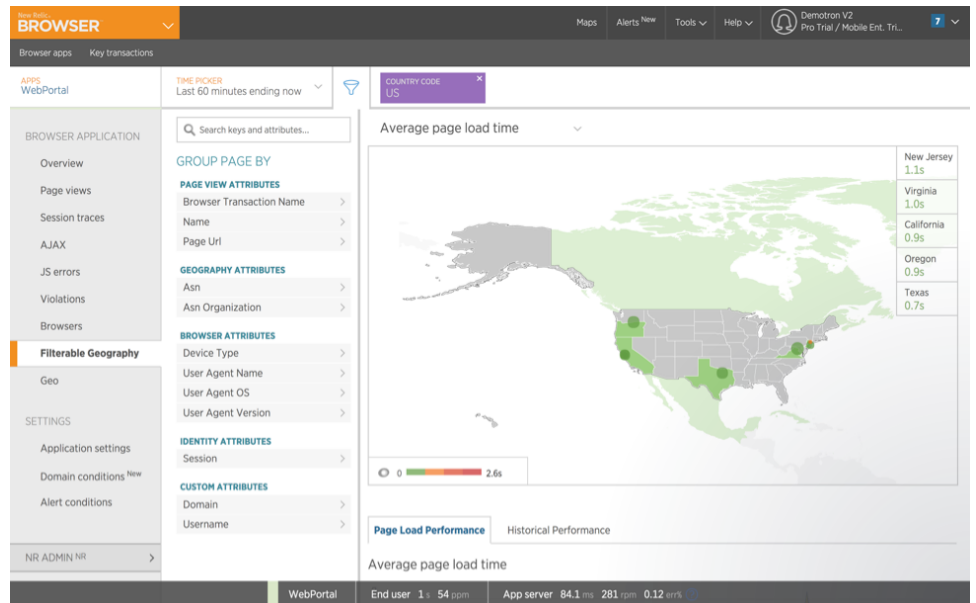


Figure 4: In addition to the Browser overview page in figure 3, you can easily compare geographic performance via the “Filterable Geography” page to easily ensure international performance baselines.

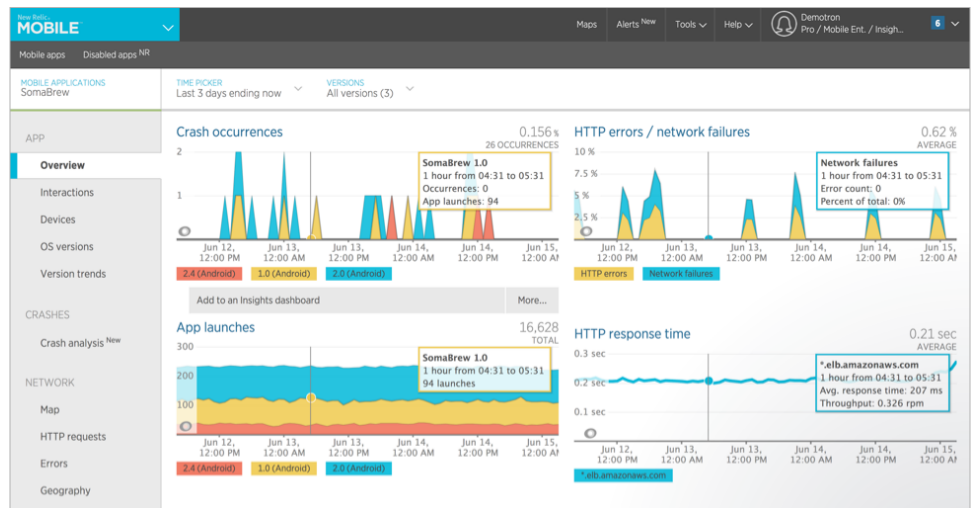


Figure 5: Mobile application performance data should be consistent across your migration but keep a close eye on HTTP response times as it indicates dependencies to backend components which are under flux.

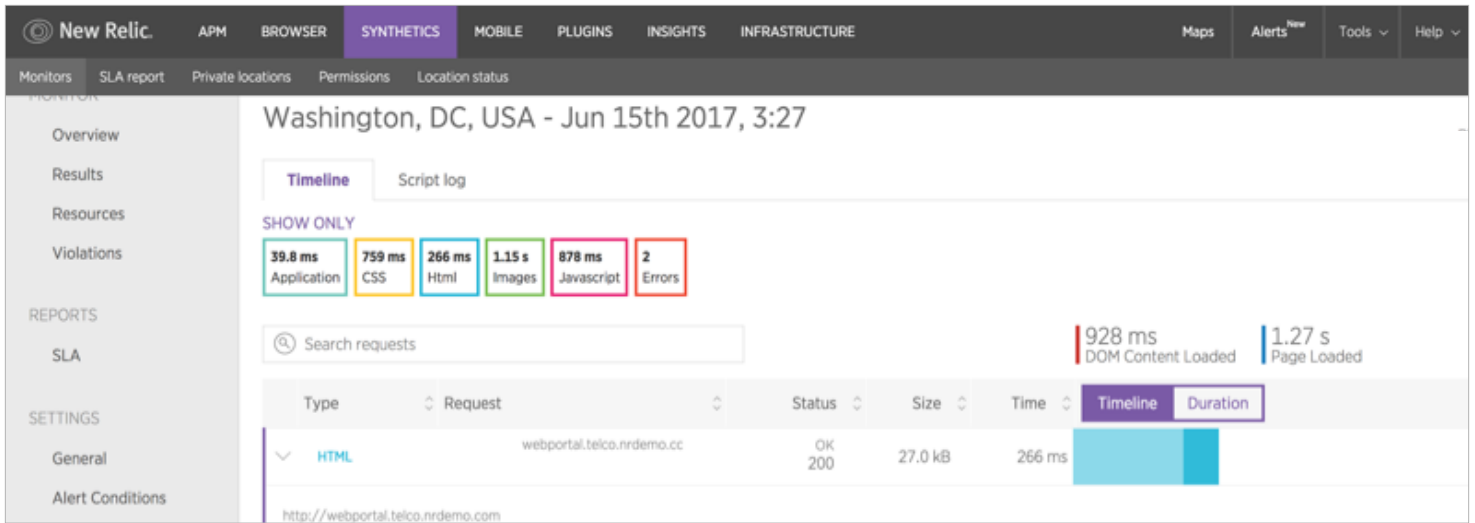


Figure 6: New Relic Synthetics allows you to drill down to the individual HTTP calls made for each asset for a Simple Browser monitor.

Backend Application Performance

While the frontend/end-user tier is the most important to properly understand whether or not you have encountered a regression during your migration, the backend applications' performance is the most integral place to understand why and how to fix those issues.

With New Relic APM, you can measure performance of your applications before, during, and after migration (see Figure 7) by providing data on:

- Response time
- Error rates
- Types of errors
- Latency
- Call rates

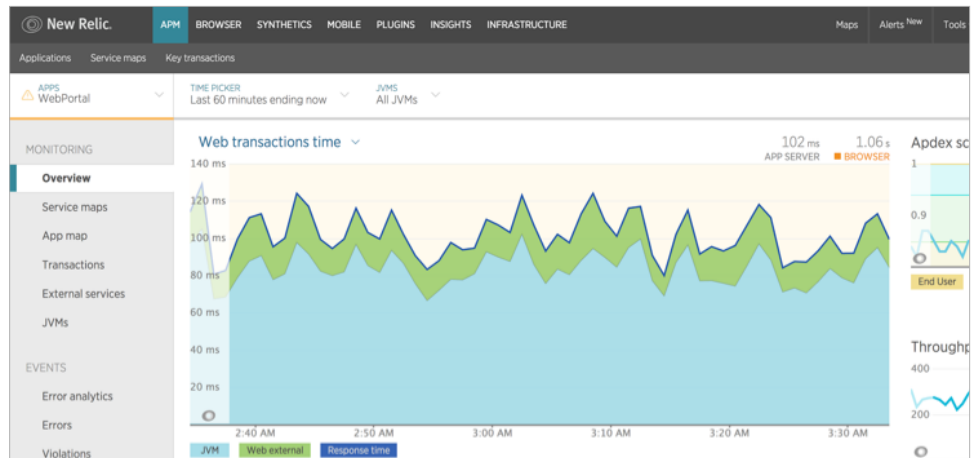


Figure 7: Easily triage individual services by comparing response times, error rates and throughput to historical norms.

If you're simply doing a "Lift and Shift" migration, you'll want to pay close attention to the latency between your services as well as the error rates of each individual service since you're adding a lot of churn to the likely fragile relationships between your applications. Instrumenting your on-premises equivalents will enable you to easily attribute issues to your migration since you're not changing your business logic. Additionally, New Relic APM will automatically compose a 'Service Map' of dependencies between your applications and other components (see Figure 8), so you can quickly understand the scope of your migration and identify any components missed as part of your port.

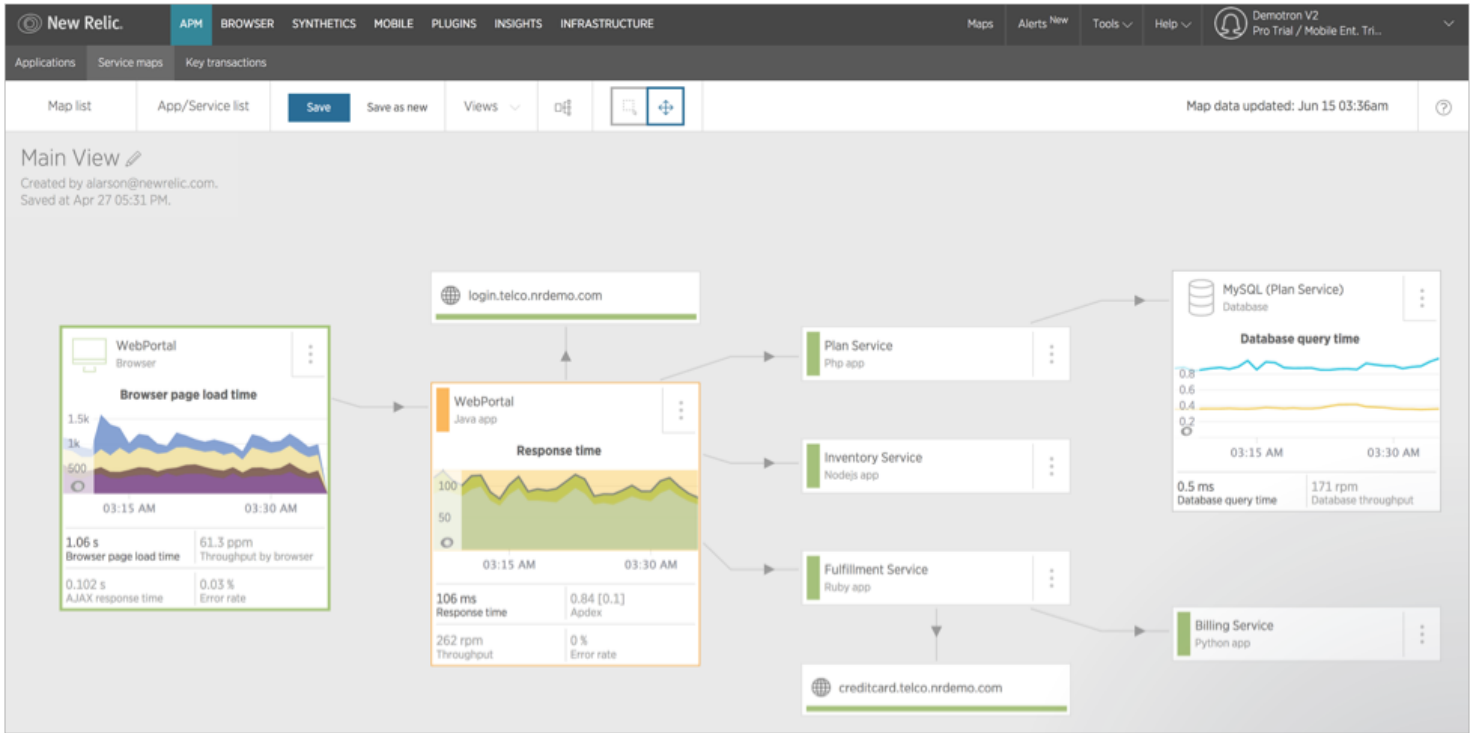


Figure 8: Discover your entire system dependency chains from browser to database even including external services with New Relic Service Maps.

Generally a latter step of your migration is to re-architect your backend applications to be more ‘cloud-oriented’. Since you are already validating your end-to-end user experience as part of the last section, you can be confident that even dramatic changes at this tier will be understood and quickly called out. When you do refactor your business logic, you can also rest assured that New Relic APM will be present to help you easily identify where your bottlenecks occur so you can quickly get to parity and better.

Infrastructure Performance

As you migrate your codebase to AWS, the majority of churn will occur at your infrastructure tier. You are literally swapping out an entire backbone for another. It may simply be different VMs in the case of “Lift and Shift” or you may be rushing toward the bleeding edge pursuing completely different architectures like “serverless” in AWS Lambda. Because of this, infrastructure monitoring of before and after is not overwhelmingly valuable. That said, when you migrate to AWS there is one important new metric that you perhaps didn’t have to care about before: cost. It doesn’t matter what hardware you used on-premises, but it certainly matters that you’re using the right hardware for your needs now that you’ve ported things over. Too small of an instance and you run the risk of degrading your application’s user experience, but too large of an instance and you’re throwing money down the drain as the machine sits idle waiting for work.

Because of this, once you have successfully migrated an application to AWS you should use New Relic Infrastructure to audit the following host-level metrics (see Figure 9) which give you a sense as to overall machine impact and whether or not you can get away with sizing down or need to size up:

- CPU
- Memory
- Load average
- Network load
- Storage
- Running processes

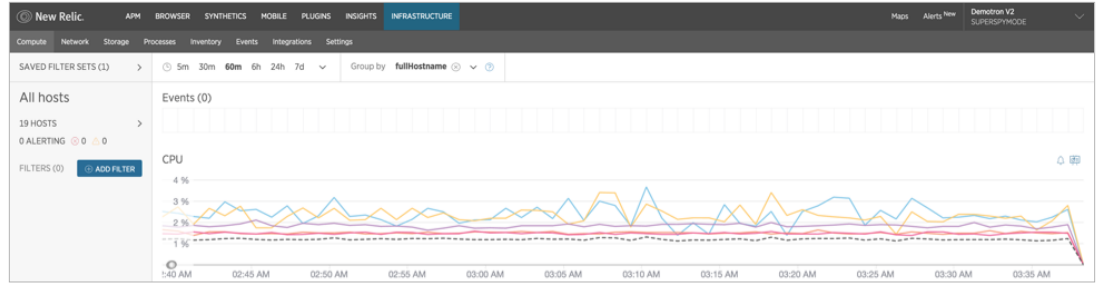


Figure 9: New Relic Infrastructure drilling into a 60 minute time window for all hosts for core KPIs.

Additionally, cloud applications generally tend to be composed of more than just a big monolithic process running on a single host. An application architected in AWS likely utilizes a multitude of their PaaS services like Elastic Load Balancer, DynamoDB or Lambda. Ignoring these components when it comes to monitoring is a bad move since they're just as important a part of your application as any code library you've chosen to use. New Relic knows this and has provided a convenient SaaS-to-SaaS integration which can be setup in minutes which will pull detailed event, metric and inventory data about a vast array of AWS services (see Figure 10) into the New Relic UI to help you ensure you have full understanding of your entire stack. Details on setting up that integration can be found [here](#).

My Baselines Are Established, Now What?

New Relic agents are installed, you've collected your initial set of metrics and studied the curated views, so what's next?

TIME PICKER

During and after migration, New Relic will continue collecting and **retaining data** for your systems. This way you can verify progress by selecting a prior date range using the New Relic time picker and hone in on any tier at an exact moment for comparison anywhere during your migration. Simply instrumenting both your on-premises and AWS-centric deploys is the easiest way to compare. Check out more details [here](#).

CUSTOM METRICS AND DASHBOARDING WITH NEW RELIC INSIGHTS

New Relic has collected a large corpus of data about your systems and provided several curated views on top of it, but sometimes the questions you need to answer are unique to your business, so New Relic created **Insights**. This product exposes the underlying data that powers New Relic's UI and allows you to craft **custom queries** that answer any questions you may have about your systems (see Figure 11).

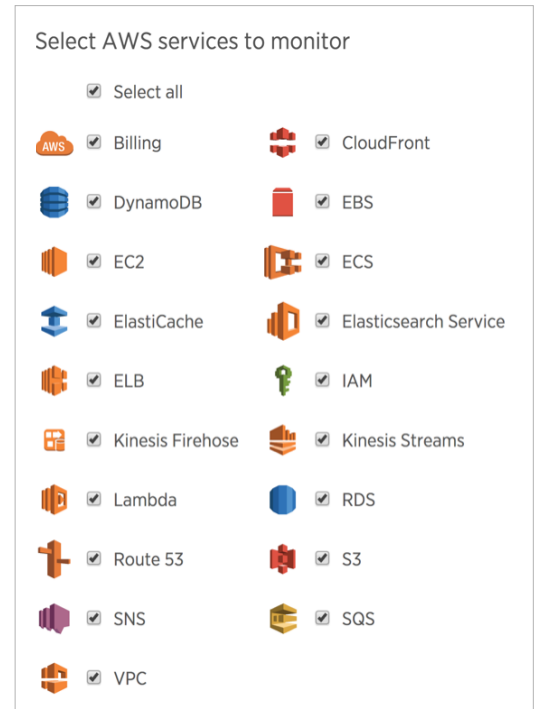


Figure 10: A list of all currently available AWS SaaS-to-SaaS integrations. Many more coming soon.

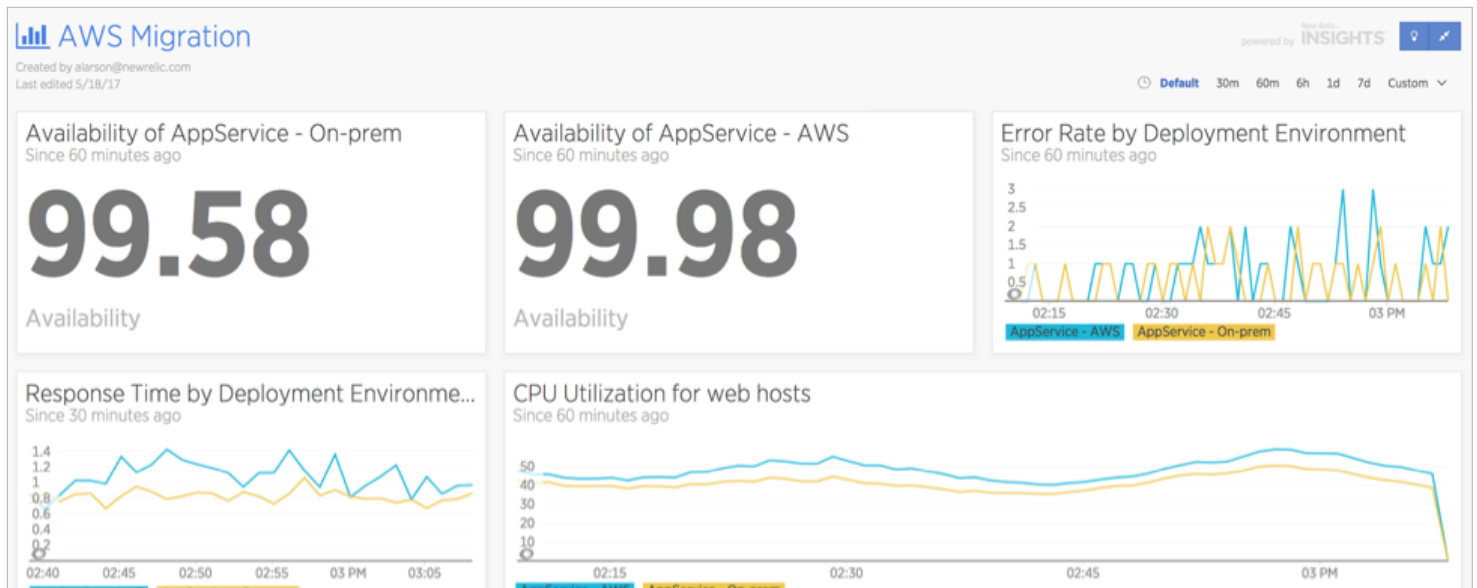


Figure 11: A very small sample of the type of completely custom dashboards you could create for comparing before and after with New Relic Insights.

Even better, you can extend existing data with **custom attributes** that are relevant to your business and view performance data through a view that is designed around what is important to your specific business. Using New Relic Insights and **labels** or just distinct names for your before and after deployments allows you to view your before and after charts on a single dashboard.

NEW RELIC ALERTS

If you are not alerting when something goes wrong, you're not really monitoring. **New Relic Alerts** will enable you to setup advanced **alert conditions** on all tiers of your application stack, so you are aware of performance degradation as fast as possible. These alerts can be associated with numerous **notification channels** that integrate directly into your existing workstreams such as ticketing systems or chat tools.

SHARING DATA

Now that you have definitive data about the success of your cloud migration, you should democratize it! There are several ways this can be accomplished:

- **Take a screenshot:** New Relic's UI aims to be intuitive and approachable by even non-technical users. Simply taking a screenshot of the interface is the easiest way to communicate with stakeholders.
- **Share a link to the page:** New Relic's interface provides a link in the footer to establish a permalink which will lock the time window via URL parameters and enable sharing your exact view with any interested parties. More details can be found [here](#).
- **Export the data and message it in other tools:** Data can be exported from New Relic in multiple ways. Metric information and entity information can be retrieved from New Relic's **data out REST API** and event data can be fetched from **New Relic Insight's export API**.



Moving Forward With Confidence

Congratulations, you've made the unknowns known and now better understand the importance of baseline metrics and how to establish them. With this data you can more easily measure, prove, and accept the success of your cloud migration project. But beyond this first step into the cloud, there are many opportunities to utilize the data provided by the New Relic Digital Intelligence Platform and improve upon your future projects: re-architecture efforts, prioritization of product development, and more. Wherever your cloud journey leads you, now you have data more readily available to ensure you can make the best possible decisions for your business—and only cut once on projects in the future.