New Relic

White Paper

Cloud Done Right

The AWS Well-Architected Framework Instrumented by New Relic

Whether you're developing cloud native applications on Amazon Web Services (AWS) or modernizing existing applications and migrating them to AWS, you have many decisions to make throughout the application lifecycle. Some of the decisions will be based on opinion or experience, such as which language to use or which application programming interfaces (APIs) will be needed.

However, many of the decisions that you make—such as sizing for AWS instances or data storage type—should be based on data-driven best practices. Taking a data-led, best-practice approach helps you create a well-architected application and application environment on AWS.

That's what the AWS Well-Architected Framework (WAF) is all about: providing "a consistent approach for customers and partners to evaluate architectures and implement designs that will scale over time."

The WAF is the result of years of experience with thousands of customers creating architectures that are built to be secure, high-performing, resilient, and efficient. Based on five pillars—**operational excellence, security, reliability, performance efficiency,** and **cost optimization**—the WAF offers detailed guidance, best practices, and foundational questions to evaluate and implement architectures that will scale over time.

When instrumented with New Relic, the WAF pillars become part of an actionable observability platform for running the digital enterprise. New Relic One delivers the full visibility, context, and actionable insight from the end-user experience through the device, application, and into the infrastructure. This is essential to achieve the full benefits of each of the five pillars—a key success factor for companies that are truly customer obsessed.

With New Relic, you can understand:

- How well your applications and environment are aligned with the WAF
- Where there are opportunities to better align with the WAF
- Where there are risks in your current AWS environment and how you can mitigate them

This white paper can help you get started using New Relic to support the AWS WAF best practices. We'll review each pillar along with examples of how New Relic can be used to instrument key portions using the metrics, events, logs, and traces (M.E.L.T.) telemetry data collected from your applications, infrastructure, and AWS services.

> *"The operational excellence pillar focuses on running and monitoring systems to deliver business value, and continually improving processes and procedures. Key topics include managing and automating changes, responding to events, and defining standards to successfully manage daily operations."*
>
> **The AWS Well-Architected Framework**

Operational Excellence

This pillar includes three best practice areas: prepare, operate, and evolve. These areas help teams understand their business and customer needs and then measure the achievement of desired business outcomes. Using the best practices from AWS, together with the observability delivered using the New Relic platform, can help you drive operational excellence by providing:

- A single location for all teams to view business value and outcomes
- Measurement and communication of the impact of changes
- Automation of issue resolution
- Valuable historical data for an incident review

Best practice example: Create a KPI dashboard

A key component of monitoring systems that deliver business value is having a single place to view key performance indicators (KPIs) related to the various aspects of your application. KPIs can include metrics such as the number of active users; which countries users are located in; the number of current orders; average order value; and users and revenue that are at risk. The example KPI dashboard in Figure 1 features charts that show these KPIs.

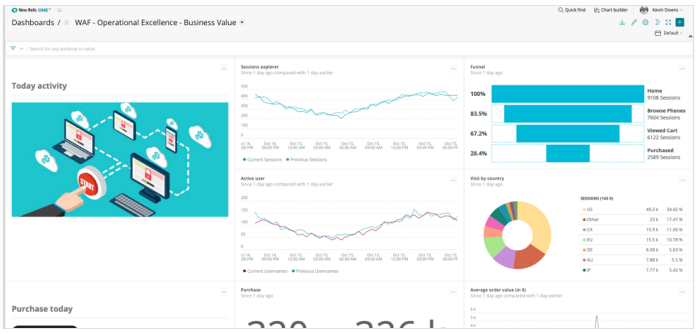


Figure 1. User, order, and risk are critical business value KPIs shown on the KPI dashboard

For managing and automating changes, integrate measurement into your development process by adding deployment markers. A design principle from the WAF recommends making frequent, small, and easily reversible changes. Proper instrumentation gives your teams full visibility into the impact of these changes on the system. Capturing tangible, measurable metrics from before, during, and after each change allows your teams to optimize changes in isolation and reduce the impact to other work happening in the system (see Figure 2).

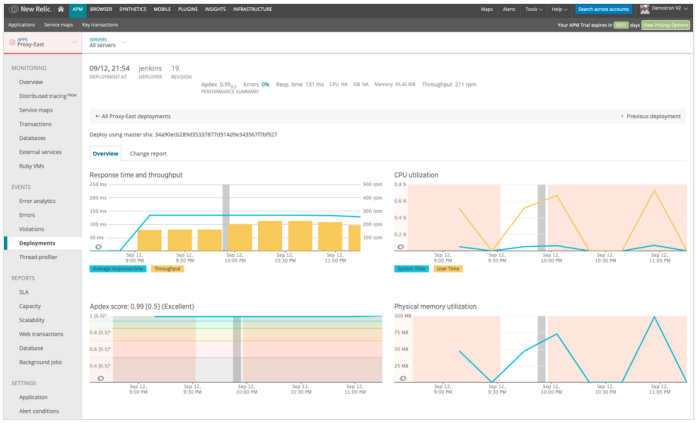


Figure 2. A deployment marker sent into New Relic through an integration with Jenkins shows measurable metrics from before and after each change.

It's also important to track deployments and the impact that code and infrastructure changes have on your end-user experience. Tracking deployments is a valuable way to determine the root cause of immediate, long-term, or gradual degradations in your application. This is one way that New Relic incorporates and extends the WAF with application and customer experience information.

Best practice example: Automate resolution

Whether you use Amazon CloudWatch alerts or choose to locate your events in the New Relic platform, it is important to automate the resolution. Automation of simple or repeatable incident response tasks

will increase efficiency and minimize the impact of incidents. With proper automation in place, you can disable or isolate faulty application components as soon as an alert threshold is reached, rather than after a notification has been issued.

For example, a team managing an application for a digital media company wants to be able to remove commenting abilities from the website if the commenting service has errors. In this case, it could:

1. Add an endpoint to their frontend web application that will toggle a feature flag enabling or disabling the UI components associated with posting comments on an article.
2. Create an alert policy with a threshold set on the sustained error rate in the commenting service.
3. Assign a webhook notification channel that will send a POST request to this endpoint, as well as to the standard team notification channels.

In this scenario, errors in the commenting system will trigger the webhook and remove the commenting UI from the website. Users can still use core functionality of the site without seeing errors generated by the commenting service. The application will maintain a stable but degraded state, allowing the team to focus on recovery without the pressure of preventing users from accessing the site.

Best practice example: Closed loop feedback

Another best practice is closing the loop after the event is over. A design principle from WAF states that you should learn from all operational failures. It is recommended that after the incident has been resolved, key stakeholders and participants capture accurate and thorough documentation of the incident to establish a review.

At a minimum, it is recommended that the documentation includes:

- Root cause analysis
- Chronology and summary of remediation steps and their result, whether they were successful or not
- Measurement of the impact to the business in terms of user experience and financial losses if possible
- Measurements of the engineering and operations time spent resolving incident
- Recommendations for system or feature improvements to prevent a recurrence
- Recommendations for process and communication improvements

Store post-mortem reports (see Figure 3 as an example) in a highly visible, searchable repository, such as a shared drive folder or wiki. It's essential that this process focuses on constructive learning and improvement rather than punishment or blame.

Post mortem	Comments
Date	March 1, 2018
Executive summary	From approximately 1:45PM until 2:30PM, users could not add items to their carts, which prevented any checkouts from occurring during the incident period.
Root cause	We determined that a change was made to the CSS rules on the product detail page that effectively disabled the Add to cart button.
Timeline	• 1:50PM: Successful checkouts < 10 for 5 minutes alert triggered; assigned to Alice. • 1:55PM: After reviewing the ecommerce team dashboard, Alice determined that the threshold was breached immediately following a deploy by Bob. She notified him of the incident. • 2:00PM: Alice and Bob began troubleshooting. Attempts at recreating the issue in production were successful. • 2:20PM: Bob determined that his change to the CSS on the product detail page disabled the Add to cart button. He deployed a hotfix. • 2:30PM: Functionality was restored and the incident was resolved.
Impact	No checkouts were completed during the duration of the incident. Our typical revenue for a Thursday during this timeframe is $30,000.
Recommendations	We have been discussing implementing New Relic Synthetics for awhile now. If we had a Synthetic check on the checkout process, this issue would have been detected immediately. We should also implement more thorough unit tests in the front-end web app.

Figure 3. Example post-mortem report

"The security pillar focuses on protecting information & systems. Key topics include confidentiality and integrity of data, identifying and managing who can do what with privilege management, protecting systems, and establishing controls to detect security events."

The AWS Well-Architected Framework

Security

There are five best practice areas within this pillar—identity and access management, detective controls, infrastructure protection, data protection, and incident response. While the best practices are primarily focused on how your organization should be using AWS security features and services (as well as third-party security solutions available through the AWS Marketplace) to protect your data and systems, New Relic can help you address three important security principles:

1. respond quickly to security events;
2. monitor and track security alerts; and
3. reduce security exposure.

New Relic can alert on any application event or forwarded log entry that has a security context.

Best practice example: Detect security events

To detect security events, New Relic recommends two main strategies.

First, build security notifications into your applications and send those notifications to New Relic in the form of logs or custom events. New Relic Logs offers a fast, scalable log-management platform that allows you to connect your log data with the rest of your telemetry data. This enables your security notification to be viewed in context with your application (see Figure 4).

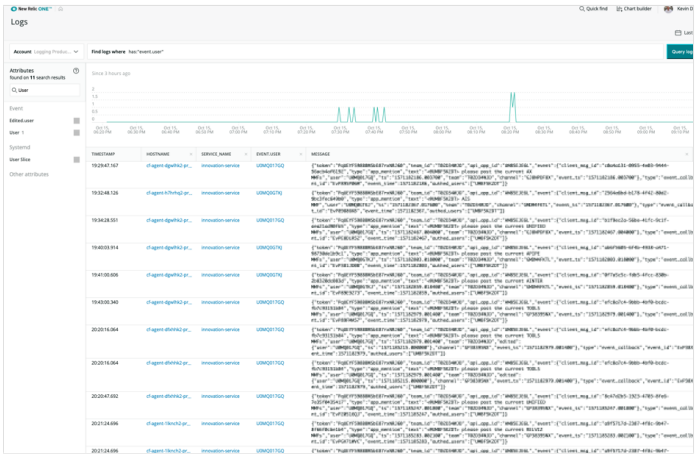


Figure 4. Log data displayed in New Relic One (image pixelated for security).

Second, integrate New Relic with both AWS and any third-party security service your company may use. Integrating with AWS CloudTrail will allow AWS account activity to be captured and displayed in context with other security information, including events related to AWS Identity and Access Management, or IAM (see Figure 5).

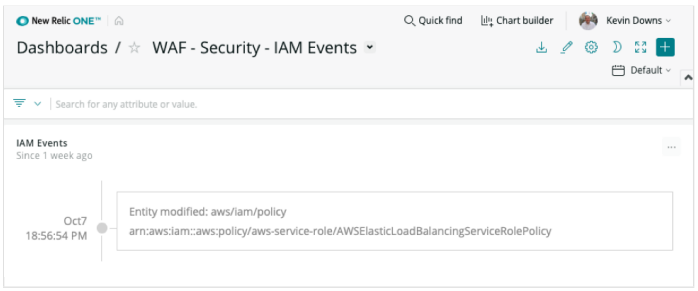


Figure 5. IAM events displayed in a New Relic dashboard.

Best practice example: Monitor security alerts

Another aspect is monitoring and reporting on potential security risks. Security indicators important to include are operating systems and their versions, as well as the Amazon Machine Images (AMIs) used. Your 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.

Armed with this information, you can receive guidance from your security team about which hosts contain any security issues. To easily find the hosts that need to be updated, use the filter ability within New Relic (see Figure 6). If your hosts are in the AWS cloud, you can filter by the additional tags available with the New Relic integration, including instance type, region, or custom tags such as application or state.

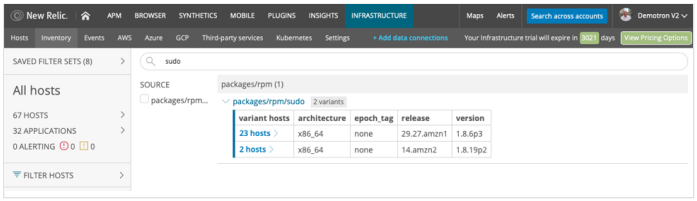


Figure 6. New Relic Infrastructure showing two hosts that contain a version of a package that includes a security threat.

Best practice example: Reduce your security exposure

One way you can reduce your security exposure is by taking advantage of AWS services to reduce the number of technologies your team must self-manage. You can do this by choosing to use managed services for new applications or replatforming existing applications to a managed service. For example, you can replatform your database to a managed service, such as Amazon Aurora, where many security tasks are handled for you, allowing you to focus on other areas. In this way, you transfer the responsibility for the management of services to AWS instead.

> *In order to better understand your role in securing your AWS environment, review the AWS Shared Responsibility Model, which outlines what you are responsible for versus what AWS is.*

New Relic provides the ability to show all of the components of an application, including potential candidates for replatforming to AWS maintained services (see Figure 7).

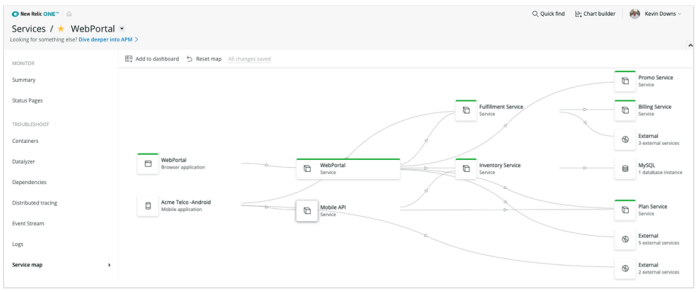


Figure 7. Service map showing the components that make up an application.

Choosing AWS managed services reduces the amount of infrastructure your organization is responsible for in regards to vulnerability and patch management. You still need to provide proper rules to protect the data and to control who has access; however, the lower-level security concerns, such as operating system patches and updates, are managed by AWS.

Another important metric to understand is how much your organization is reducing the security surface area and effort by using managed services. How many databases do you no longer need to track, manage, and patch from a security perspective? In addition to the number of databases, how many types of databases (such as MySQL, Oracle, Microsoft SQL Server) and how many different ver-

sions (1.2, 2.7, 5.4.2, etc.)? The more database types and versions you need to secure, the more your security challenge increases.

> *"The reliability pillar focuses on the ability to prevent, and quickly recover from failures to meet business and customer demand. Key topics include foundational elements around setup, cross project requirements, recovery planning, and how we handle change."*
>
> **The AWS Well-Architected Framework**

Reliability

The Reliability pillar includes three main best practice areas: foundations, change management, and failure management. With data from New Relic, you can create a reliable system, one that is stable, predictable, and highly available. You can use New Relic to help find errors early in development, prevent failures, and quickly recover from any that do occur, with the goal of delivering a great customer experience.

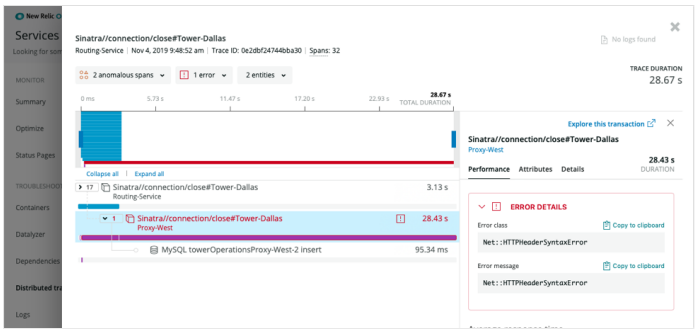
Best practice example: Find errors early

A best practice for creating reliable applications and preventing failures in production is to find errors as early as possible by using continuous integration and continuous delivery (CI/CD) concepts.

Continuous integration refers to developers maintaining strict control over code compatibility. In practical terms, it's the process of automating the build and testing of code every time a developer makes changes. Continuous integration, however, takes enterprises only part of the way to quickly putting high-quality software into users' hands. The next step in that path: continuous delivery.

Developers who practice continuous delivery produce code that is always deployable and ready to go into production. Continuous delivery is a collection of software development practices and methodologies that accelerate time to market while improving quality.

Modern applications and sites increasingly use many interconnected services. An application architecture that relies on many services or microservices is often referred to as a distributed system. Using distributed tracing in the New Relic platform allows for the tracking of the activity resulting from a request to an application. Being able to trace the path of a request as it travels across a complex system will help you discover any latency with the components along the path. Being able to track latency to the component allows you to find bottlenecks in your applications. Bottlenecks in your application usually point to an error. The earlier you find an error, the faster it can be corrected and the overall performance and reliability of your application improved.



Find errors fast using distributed tracing. Here we see an HTTPHeaderSyntaxError causing a 28 second delay.

Best practice example: Track uptime against SLAs

Today's digital businesses have stringent uptime requirements for critical applications. Start with a clear understanding of your service level agreement (SLA) and uptime requirements. Then compare reliability metrics to see how the application is performing against the SLA. What is your rate of application availability? How often are your users impacted by errors? New Relic can help you monitor your KPIs and SLAs around uptime (see Figure 8).

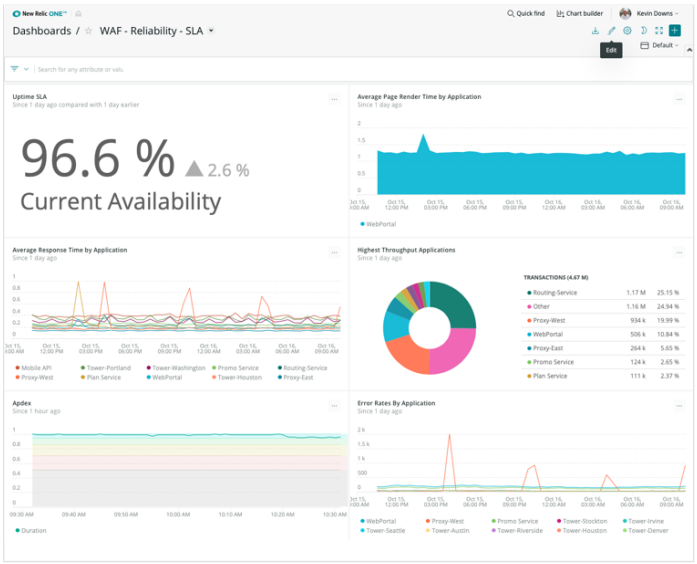


Figure 8. Monitoring SLAs is a best practice to understand application reliability.

Best practice example: Understand scaling performance

For many applications, it's difficult to anticipate and predict requirements for scaling into the future, even if it's only a few months in advance. If your application is planned to support only a small number of users, you don't need to worry much about how well the application scales. However, the expectation for most applications is that it should support an increasingly higher number of users. To make sure your application can support an increase (or decrease) in users, scaling must happen automatically. You shouldn't be spending time tuning your autoscaling rules. With the New Relic integration to AWS Auto Scaling, you get metric and inventory data about your Auto Scaling groups (see Figure 9).

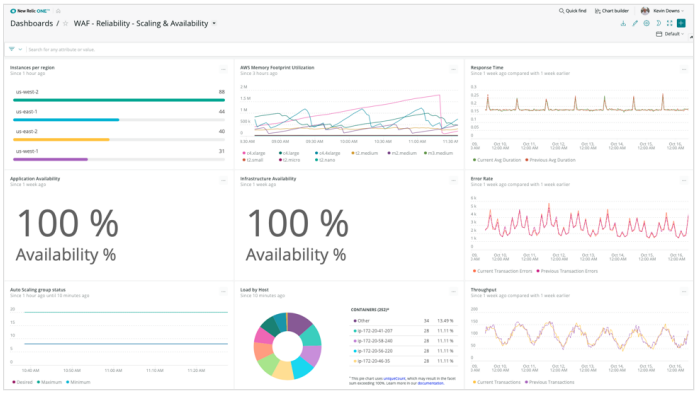


Figure 9. Scaling and availability information on your application.

> *"The performance efficiency pillar focuses on using IT and computing resources efficiently. Key topics include selecting the right resource types and sizes based on workload requirements, monitoring performance, and making informed decisions to maintain efficiency as business needs evolve."*
>
> **The AWS Well-Architected Framework**

Performance Efficiency

This pillar is about using computing resources efficiently to meet system requirements and maintain efficiency as demand changes and technologies evolve. The four best practice areas in this pillar—selection, review, monitoring, and

tradeoffs—each rely on data to help you select and evolve the right infrastructure and technologies to maintain performance efficiency.

New Relic lets you instrument everything so you have no blind spots when it comes to performance efficiency in AWS. Baseline your current environment, then make informed decisions on where to rightsize your infrastructure. Use real user KPIs to determine where, geographically, to place your cloud resources to achieve the best performance. Finally, monitor new cloud technologies (e.g., Amazon EKS and AWS Lambda) as you experiment with them to deliver better outcomes.

Best practice example: Make informed decisions

To optimize how your applications use resources, start by collecting baseline performance data in the AWS environment using New Relic. By understanding your current state, you can make informed decisions on how and where to focus improvement efforts.

The AWS cloud offers many advanced technologies to help customers achieve efficient performance to host their applications, including Amazon Elastic Compute Cloud (EC2); AWS Lambda for functions-as-a-service; Amazon Elastic Kubernetes Service (EKS); and the many databases available within Amazon Relational Database Service (RDS). These cloud services, properly sized and monitored, can meet application system requirements as customer demand changes and technologies evolve.

Best practice example: Rightsize your environment

New Relic can help you understand how applications impact infrastructure performance and provide a valuable data-driven approach to rightsizing your environment. In the example in Figure 10, it's clear that both the c4.large and c4.xlarge are underutilized for CPU. This would indicate that using a lower-tier C (compute) instance would be advised.

However, memory utilization for both of these instances ranges between 80% and 20% over time. By reviewing both CPU and memory information, you would switch these instances to a memory-optimized instance. In this case, by rightsizing for memory, the CPU utilization will also increase, making the performance of this application more efficient. (Note: The chart on the right in Fig. 10 will allow you to observe how the customer experience is impacted as you strive to rightsize.)

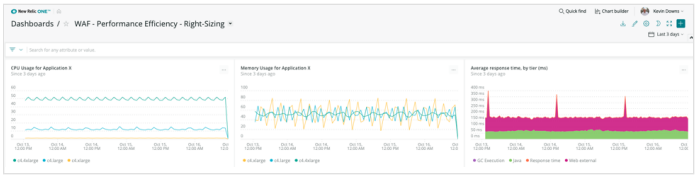


Figure 10. Rightsizing your environment requires taking multiple KPIs into account.

Best practice example: Optimize for user location

Another performance 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. You can take advantage of geographies that allow you to more efficiently serve your customers by locating resources based on user location and business goals.

In this next example, let's assume the target customer is in Columbus, Ohio. By observing the customer journey from the homepage through checkout, you will have a complete window into the KPIs that demonstrate how well your application is performing, as shown in Figure 11.

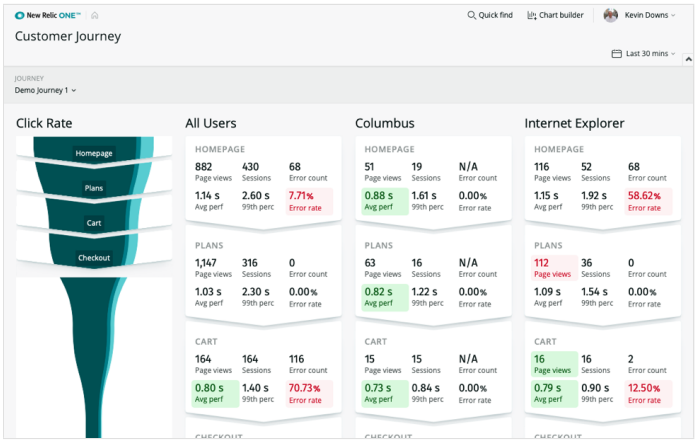


Figure 11. Performance monitoring showing target GEO.

Best practice example: Experiment with new technologies

One of the benefits of AWS is the ability to experiment more often. Trying new technologies to achieve better overall performance can be beneficial for both you and your customers.

One growing new technology area that many organizations have begun using is containers and Kubernetes, a mission-critical platform for managing containerized workloads. New Relic offers a Kubernetes cluster explorer that provides a multi-dimensional representation of a Kubernetes cluster, letting you zoom into your namespaces, deployments, nodes, pods, containers, and applications. With the cluster explorer, you can easily retrieve the data and metadata about these elements to understand how they are related.

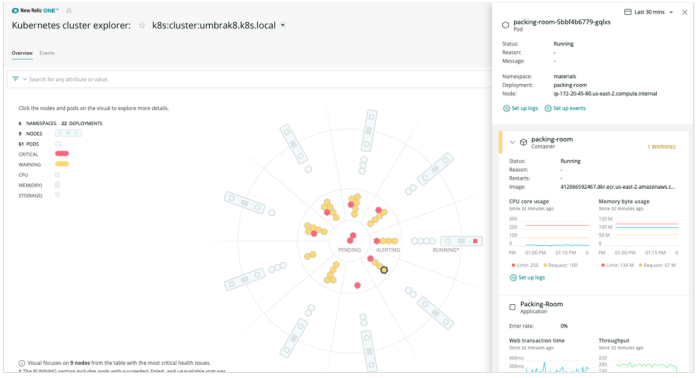


Figure 12. New Relic Kubernetes cluster explorer displaying KPIs for a specific pod.

Another newer technology is the concept of function-as-a-service (FaaS), such as AWS Lambda. Lambda offers the ability to scale without the need to allocate additional resources—AWS handles this for you. This gives companies that experience sudden spikes a performance advantage.

New Relic offers performance monitoring for your serverless AWS Lambda functions. This is extensive Lambda monitoring that uses both CloudWatch data and code-level instrumentation to deliver a more in-depth monitoring experience (see Figure 13). New Relic's Lambda monitoring gives you visibility into:

- Every invocation of your Lambda functions, including performance data such as duration, memory usage, cold starts, exceptions, and tracebacks
- Invocations for other AWS services, such as publishing messages in a Simple Notification Service (SNS) topic, or items placed in a DynamoDB table, and the operation and target for those services
- Request paths that led to your Lambda via distributed tracing, and how Lambda spans across other distributed traces in your environment
- Information about the source that triggered the Lambda

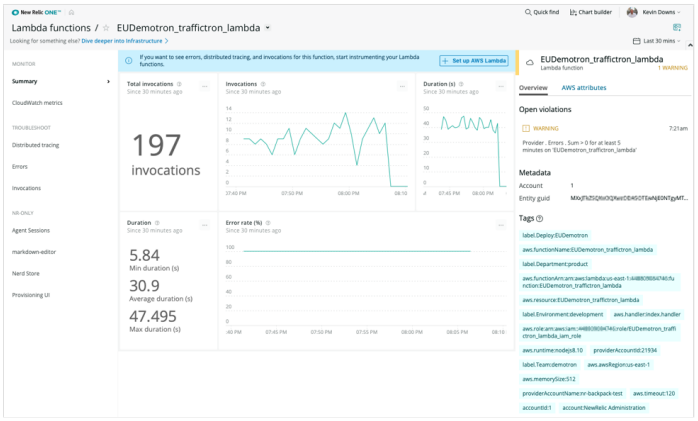


Figure 13. New Relic monitoring for AWS Lambda.

New Relic.

> *"Cost Optimization focuses on avoiding un-needed costs. Key topics include understanding and controlling where money is being spent, selecting the most appropriate and right number of resource types, analyzing spend over time, and scaling to meet business needs without overspending."*
>
> **The AWS Well-Architected Framework**

Cost Optimization

New Relic takes the cost optimization pillar to the next level, helping you apply the four best practice areas of expenditure awareness, cost-effective resources, matching supply and demand, and optimizing over time. Because cost optimization is not simply about the amount that is spent, but the impact of the application spend, New Relic lets you incorporate business data via customer-specific events to track outcomes such as revenue and performance impact by geography.

Best practice example: Match investment to objectives

For example, performance in Chile might not be at the level you want it to be; however, it wouldn't make sense to invest in that geography currently because it isn't one of your top targets. China might have very few end users currently, but your organization wants to invest heavily to grow business in that country. Therefore, you need to prioritize the experience of the users there. This combination of performance data and company objectives should influence how you determine appropriate investments.

Best practice example: Track user satisfaction

Another way to see if you're optimizing costs is to review your application's Apdex score in New Relic. Apdex is an industry standard to measure user satisfaction with the response time of web applications and services, which helps you see how satisfied users are with your application. You can use New Relic to compare your Apdex to the costs incurred by the application. Is this an appropriate level of investment for the role the application plays? What should your targets be for the end-user experience? Your goal is to balance these metrics with the costs, allowing you to meet your end-user experience objectives without breaking the bank.

In the example in Figure 14, the Apdex and availability are 100% and customers are experiencing very fast page load times. However, if you look at the utilization of the infrastructure and database, as well as the queue length, you will see that they are very low. This is because the cloud infrastructure environment has been over-provisioned. If you rightsize these components, you can achieve a better balance of your cloud spend and end user experience.

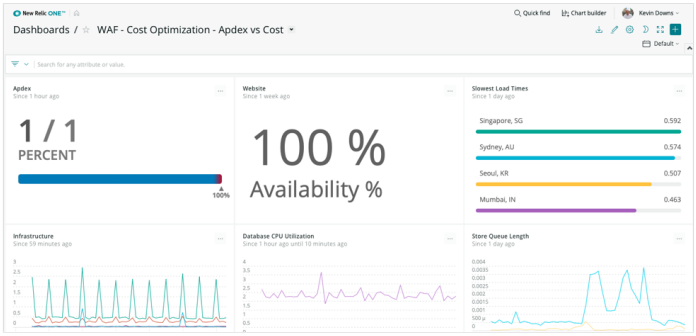


Figure 14. Apdex versus cost helps you optimize your infrastructure to match desired spend and performance.

New Relic also offers a Cloud Optimize tool. Using this tool will quickly point you to instances that are over- or under-provisioned (see Figure 15).

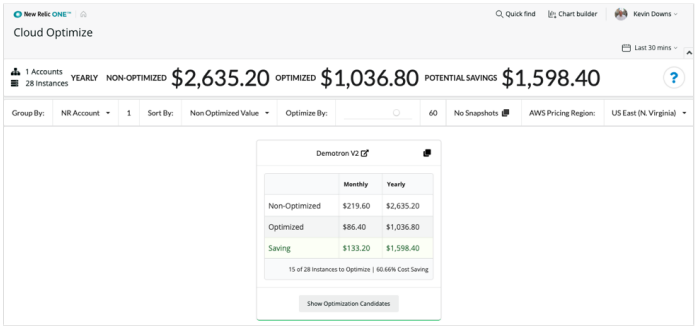


Figure 15. New Relic Cloud Optimize tool showing potential savings.

Conclusion

Using the AWS Well-Architected Framework to guide you as you make decisions about your cloud environment can help ensure you are designing a reliable, scalable, secure, efficient, and cost-effective environment for your new and modernized applications. The resulting environment helps your DevOps teams produce and operate stable and efficient systems that deliver the business outcomes your organization wants to achieve.

When instrumented by New Relic, the best practices within the WAF can be become measurable and observable within your digital enterprise, helping you deliver the high-quality applications that your customers demand.

Get more from your AWS investment

With observability for every stage of your modernization journey, you can unlock the full benefits of the cloud. Visit newrelic.com.