**A framework of critical DevOps interview questions**

To give your DevOps interview prep a serious head start, we built a framework of critical DevOps interview questions and invaluable insights on crafting answers to them.

First, we talked to a pair of New Relic pros with deep experience building and working in DevOps teams: solution consultant [Eric Mittelhammer](https://www.linkedin.com/in/ericmittelhammer) and software engineer [Beth Adele Long](https://blog.newrelic.com/author/bethlong/). We also connected with Jim Johnson, senior vice president at national recruiting firm [Robert Half Technology](https://www.roberthalf.com/work-with-us/our-services/technology) to find out the kinds of questions companies are currently asking DevOps candidates during phone and face-to-face interviews.

**Q: How would you describe the functions of an ideal DevOps team?**

Right off the bat, here’s an interview prompt that, as Eric notes, intersects all facets of DevOps. It’s almost inevitable that you’ll face one or more “big picture” questions like this to test your understanding of DevOps and to show that you see it as more than a set of tools or a development methodology.

This is a rather open-ended question, Eric says, and the answer can include points and anecdotes about processes, tools, and culture. But they should all center around the theme of removing obstacles that slow the pace at which code is delivered to end users by using the right tools to give you the confidence to do it.

In answering, you’ll want to hit themes like:

* Reducing the boundaries between those who own software and those who own hardware
* Empowering teams to make decisions about the tools and infrastructure they use
* Encouraging small, frequent code releases that address as narrow a scope of functionality as possible (small changes are easier to roll back and fix)
* Eliminating testing and QA environments
* Releasing directly to production and relying on instrumentation and performance metrics to validate releases

**Q: How would you describe DevOps?**

Here’s Jim’s—our recruiter—perspective for developing a strong response to this question:

*“Make sure you highlight the main goal of any DevOps project, which is to improve collaboration in planning and delivery to achieve faster deployment of a project, service, or software to create business value. Dive further into your answer by listing some technical and collaborative examples of DevOps projects you’ve worked on, or perhaps studied, in the past. The employer wants to know your familiarity with the DevOps process and gauge your experience level.”*

The bottom line with these big-picture, high-level questions about DevOps is simply to be prepared for them. On the surface, they seem straightforward, but if you haven’t developed a cogent response, they could leave you hemming and hawing during the interview, which is never a good position to be in.

Jim recommends leaning significantly on your past experience. (More on that in a moment.) If you’re just starting out, though, you can study up on DevOps success stories, as there are plenty floating around. Media giant [Gannett, for example, is a great case study in DevOps success](https://blog.newrelic.com/technology/modern-software-podcast-gannett-kubernetes-sre/).

**Related links:**

* [**Case study: Etsy, Sprouter and Conway’s Law**](https://itrevolution.com/etsy-sprouter-and-conways-law/)**(IT Revolution)**
* [**Case study: What the enterprise can learn from Etsy’s DevOps strategy**](https://www.computerweekly.com/news/4500247782/Case-study-What-the-enterprise-can-learn-from-Etsys-DevOps-strategy)**(ComputerWeekly.com)**

**Q: Can you describe a previous success within your DevOps experience?**

Being able to illustrate your big-picture understanding of DevOps with a particular project or team experience is incredibly useful in an interview setting. It lets the hiring team know that you’re the real deal, rather than someone who’s just littering their resume with a bunch of DevOps-oriented catchphrases and hoping for the best.

Even if you haven’t had significant real-world DevOps experience, try to put your past experience in the best light, and ensure that you can connect your individual contributions to the broader team and business:

“Bring up projects in which you’ve been a key player,” Jim says, “How did you help your team succeed? What technology and processes did you use? What value did this project bring to the business? Touch on the project as a whole, but also emphasize your main contributions and how your team worked together.”

**Q: What skills have you learned to help you better succeed in a DevOps role? How did they help?**

At just about any career stage or experience level, hiring managers in DevOps environments tend to look for people who show a commitment to continuous learning and skills development. Tool *X* might be the best solution for a particular process or problem today, but there’s no guarantee it will still be right for a problem that arises tomorrow. Take the time to reflect on how you’ve added new skills and experience over time, whether it was learning a new programming language, building a sandbox project to learn a new cloud platform, or taking advantage of educational opportunities offered by a previous employer.

Jim says: “Be prepared to talk about any certifications or skills you’ve gained, why you did so, and any results that may have come from it. The employer will want to know how dedicated you are to your professional development. It also shows that you went above and beyond to meet a challenge or fill a need for the business.”

**Q: Explain why proper instrumentation and measurement are so integral to DevOps culture and process. How do you use them in a development and deployment workflow?**

This is another great example of the need to be able to connect the dots between the technical aspects of DevOps and equally important pieces like culture and measurement.

For advice on answering such questions, Eric says it’s important to know that “DevOps relies heavily on automation and tools rather than people and process. Properly instrumented systems will let us know when they are under stress, rather than relying on humans such as testers—or even worse, end users—to discover those conditions. Candidates should be able to describe a workflow in which they make small, atomic, incremental changes. Their deployment process should be as automated as possible, and they should use instrumentation tools to verify the impact and correctness of their change.”

*Don’t miss our post*[*Guide to Measuring DevOps Success*](https://blog.newrelic.com/2018/07/10/measure-devops-success/)

**Q: How do you define observability, and what advice would you offer an organization working to improve the observability of its systems?**

According to Beth Long, some organizations will introduce the concept of observability into a conversation around measurement and instrumentation. While the use of a single term may not seem like a game-changing distinction, Long notes that it provides some very important insights into a team’s DevOps practices and culture.

“Observability is a measure of your ability to understand what’s really happening inside a system,” Long says. “It’s a critical concept today, as we deal with increasingly complex, distributed, and often ephemeral application environments where it’s often very difficult to understand the properties of an application and its performance.”

According to Long, how you go about instrumenting a system can have a huge impact on your ability to monitor and gain visibility into it. “You instrument and monitor a system as part of a broader strategy to make the system more observable,” she explains.

Long recommends that a candidate break observability down into two issues, both of which tie into DevOps practices and priorities: “First, you need to understand what types of data flow from an environment, and which of those data types are relevant and useful to your observability goals,” Long says. “Logs, traces, business analytics, data from third-party tools—all of it needs to be assessed and integrated.”

Second, Long states, it’s necessary to get a clear vision of what a team cares about. “What’s our strategy for making sense of all this data—distilling, curating, transforming it into actionable insights into the health and performance of your systems.”

Finally, Long points out that questions about observability offer potentially useful clues about an organization’s DevOps maturity level.

“Observability as a concept has been around for decades—it’s far from new,” she says. “But as a concept for working with modern application environments and for getting value from measurement and instrumentation, it’s a relatively recent development—and it suggests that an organization is at least paying attention to what’s happening on the cutting edge.”

**Learn more about**[**Observability and Instrumentation: What They Are and Why They Matter**](https://blog.newrelic.com/technology/observability-instrumentation/)

**Q: If there is a new technology or process you’d recommend for improving our DevOps strategy, how would you evaluate said improvement?**

Candidates may overlook the importance of *measurement* to DevOps, thinking it’s something to consider *after* they get the job. But a high-functioning DevOps team is definitely going to want to gauge how you measure the efficacy of tools and processes.

Here’s Jim’s advice on developing a good response to this question:

*“DevOps transformation relies heavily on a team’s ability to implement new tech and processes. You should explain how you go about researching, evaluating, and trying out new technologies and strategies, and how you would explain to stakeholders how this will improve the project and create value for the business.”*

It’s pretty likely the interviewer is going to want to hear about how you’d evangelize the value of DevOps to the broader organization when necessary; they’re also ensuring that when it comes to things like selecting a new tool, you’ll be more thoughtful than “throwing some spaghetti against the wall to see if it sticks.”

**Q: What does “infrastructure as code” mean, and how does that idea fit into DevOps culture?**

Automation is one of the fundamental concepts of DevOps—it’s what makes the promised speed, agility, and efficiency possible. For many organizations, thinking of infrastructure as code is the key their overall automation strategy. As Eric explains:

*“Infrastructure as Code is the practice of managing infrastructure with a single ‘source of truth’ such as a configuration or source code file that can be versioned and used by automated systems to create, provision, and configure infrastructure with minimal human involvement. This greatly increases the speed at which new systems can be provisioned and existing systems can be expanded, scaled, and improved. Furthermore, it allows the team developing the software to specify exactly what they need, without necessarily having to know how to provision it themselves.”*

Depending on the organization and its job description, DevOps candidates may also want to bone up on infrastructure automation tools and container technology:

* [**The Best Tools for Cloud Infrastructure Automation**](https://blog.newrelic.com/2018/04/19/best-cloud-infrastructure-automation-tools/)
* [**What Is Container Orchestration?**](https://blog.newrelic.com/2018/07/17/container-orchestration-explained/)
* [**Docker vs. Kubernetes: It’s Not About One or the Other**](https://blog.newrelic.com/2018/06/13/docker-vs-kubernetes/)

**Q: How would you go about diagnosing and fixing problems in production?**

You should expect questions on process and practice; the interviewer will want to know how you go about solving problems in your day-to-day job. Eric advises focusing on the importance of metrics and using the proper tools to monitor the metrics that matter most to the organization, so you can act when necessary:

*“Candidates should talk about which performance metrics are important to them and why. They should discuss both user-focused timing and latency metrics, such as response time and [Apdex score](https://docs.newrelic.com/docs/apm/new-relic-apm/apdex/apdex-measuring-user-satisfaction), as well as application-wide quality metrics such as error rate and throughput. They should know how to use their tools to alert them to critical conditions on these metrics, and then describe how they can use them to drill down to find out what specifically is causing the problem.”*

**Related link:**[***DevOps Without Measurement Is A Fail***](https://newrelic.com/how-to-measure-the-success-of-devops?content=eBook)

**Q: How would you take our company’s DevOps strategy to the next level?**

If you’re going for a mid-level or senior position, the interviewing team will likely ask for your thoughts on the next phases of its DevOps transformation. This is definitely a “big-picture question,” but you’ll want to specifically connect your skills and experience to this particular position and organization. Jim suggests answering this way:

*“The hiring manager wants you to think strategically and apply your previous experience to the challenges they’re facing. It would be beneficial to explain that you understand DevOps is a culture as well as an efficiency process, creating a collaborative environment across teams. Discuss any improvements you’d make to the collaboration factor of their DevOps strategy in addition to any technologies you’d recommend to the team.”*

The importance of this question lies in the fact that successful DevOps teams—and the executives and managers responsible for building them—know that there’s no completion date for a DevOps journey. It’s about continuous evolution and improvement. You’ll want a thoughtful answer that offers specific ideas on what that continuous evolution and improvement might look like for the particular team or company interviewing you.

**Finally, what should you ask the interviewer?**

Conventional wisdom recommends that you always ask a few questions of your own in a job interview; it shows that you’re not only interested in that particular employer, but that you’re also “interviewing them.” Moreover, it shows you want to understand their particular iteration of DevOps culture, which is especially important given that there’s not really a one-size-fits-all blueprint.

When it’s your turn to ask the questions, Eric suggests the following:

**Q: How quickly do you expect new hires to begin contributing and deploying code?**

This might seem like a very specific question, but it’s a great way to tell if the employer is a DevOps shop in name only.

“Teams that are serious about DevOps will do their best to remove as much friction as possible from the development and deployment process,” Eric explains. “Ideally, a new hire should be able to commit and deploy source code on their very first day, whether that be a small bug fix, or simply adding their name to a [humans.txt](http://humanstxt.org/) file.”

If the employer indicates it’s going to be a while, you might want to probe a bit deeper; it could be sign that they’re saying “DevOps!” but that their actual processes and culture aren’t there yet.

“Their answer to this question could be considered a holistic measure of the organization’s collective and cultural trust in their development team as well as the maturity of the team’s continuous integration and continuous delivery process,” Eric says. “In other words, you’re finding out how physically easy it is to move code to production.

**Q. How committed is the organization to embracing and practicing a true DevOps culture?**

You probably don’t want to ask this question verbatim. According to Beth Long, however, it’s critical to know which you’re more likely to encounter: a healthy and balanced work environment, or a situation that may be unhealthy and even toxic in the long run.

A great way to get at this question is to ask more focused questions that gauge an organization’s commitment to DevOps cultural values. “One fairly direct question that you can ask involves how an organization protects staff against burnout,” Long says. “Are team members expected to take time off every year? Are on-call rotations and incident response practices designed to spread the load?”

In addition, she states, questions about an organization’s incident response process, and especially how it learns from incidents, can be revealing. “Is an organization committed to progressive and equitable practices, such as the use of blameless postmortems to learn from incidents?” Long asks.

Finally, Long advises simply staying aware of the depth and extent of an organization’s DevOps practices: “If a DevOps platform team is all-in on embracing the culture, but other teams seem to be doing business as usual, it’s not a deal-killer—but it’s definitely a sign that you should dig deeper into its commitment to a healthy, sustainable DevOps culture.”

*,*[*New Relic One*](https://blog.newrelic.com/tag/new-relic-one/)*,*[*observability*](https://blog.newrelic.com/tag/observability/)*,*[*Telemetry Data Platform*](https://blog.newrelic.com/tag/telemetry-data-platform/)*,*[*what's new*](https://blog.newrelic.com/tag/whats-new/)

A new season represents a time for change. As we reflect on the past few months at New Relic, we’ve had quite a few changes of our own. In July, we [simplified our pricing model](https://blog.newrelic.com/product-news/simple-pricing/) and [reimagined New Relic One](https://blog.newrelic.com/product-news/reimagined-new-relic-one-experience/). Since then, we’ve released 40+ new features, capabilities, and integrations to help you deliver value to your customers.

To recap this season, we’ve asked our engineers and product managers to list [their favorite features](https://one.newrelic.com/launcher/jerome.plg-whats-new-launcher) to help you be more productive and collaborative.

Here are their top 10 picks:

**1. Dark mode**

When you’re resolving incidents in the middle of the night or want to avoid straining your eyes, you can view New Relic One in dark mode. Dark mode is way more than just a cool feature (though it still is that). For many industries, dark mode is essential. Learn more [here](https://blog.newrelic.com/product-news/dark-mode-for-mission-critical-operations/).

**2. New AWS integrations**

Get end-to-end visibility into your AWS cloud services and the rest of your stack with valuable new integrations:

* [AWS Lambda Extensions](https://blog.newrelic.com/product-news/aws-lambda-extensions-integrations/) substantially ease your ability to send telemetry data from AWS Lambda to New Relic One.
* [AWS Control Tower](https://d1.awsstatic.com/Marketplace/solutions-center/downloads/New-Relic-AWS-Control-Tower-Implementation-Guide.pdf) automatically integrates with New Relic One for single or multi-account environments enrolled in AWS Control Tower.
* [AWS Distro for OpenTelemetry](https://blog.newrelic.com/product-news/aws-distro-for-opentelemetry/) uses the OpenTelemetry Collector and New Relic exporter to send telemetry data from your AWS services to New Relic One.
* [AWS X-Ray Integration](https://blog.newrelic.com/product-news/aws-x-ray-integration/) automatically combines traces from AWS managed services with traces from New Relic for end-to-end observability visualized entirely within New Relic One. You can capture, filter, and query it all—no manual instrumentation required.
* [Amazon Kinesis Data Firehouse](https://blog.newrelic.com/product-news/amazon-kinesis-data-firehose/) helps you ingest and forward CloudWatch Logs data into New Relic One to expand the insights into your cloud stack.
* [AWS Bottlerocket](https://blog.newrelic.com/product-news/aws-bottlerocket/) helps provide full visibility into workloads and infrastructure, including Amazon EKS and Amazon ECS in New Relic One.

New Relic also achieved [AWS Outposts Ready Designation](https://newrelic.com/press-release/20200915), which recognizes that New Relic One offers complete visibility into AWS compute, storage, database, and a full range of other available AWS services in the AWS regions.

If you build, deploy, or maintain apps and critical workloads on AWS, you can now get full, instant access to all of New Relic One for free, and only pay for what you use beyond 100 GB per month when you [subscribe to New Relic in the AWS Marketplace](https://blog.newrelic.com/product-news/free-observability-plan-aws-marketplace/).

**3. OpenTelemetry UI**

New Relic One now has a UI dedicated to providing full APM functionality for your OpenTelemetry data. Send your OpenTelemetry data to New Relic using one of the [OpenTelemetry exporters](https://docs.newrelic.com/docs/integrations/open-source-telemetry-integrations/open-source-telemetry-integration-list/new-relics-opentelemetry-integration" \t "_blank) and quickly discover and analyze your data to optimize the performance of your applications and services using one of seven key pages. Learn more [here](https://blog.newrelic.com/product-news/opentelemetry-user-experience/).

**4. New Relic Edge with Infinite Tracing**

New Relic users with Pro or Enterprise [Full-Stack Observability](https://newrelic.com/platform/full-stack-observability) can now access and benefit from [New Relic Edge](https://newrelic.com/products/edge-infinite-tracing) with a fully managed, cloud-native, tail-based distributed tracing service. This new service observes 100% of all application traces across your distributed systems, and provides visualization and storage for the most actionable data so you can investigate and solve issues faster. Learn more [here](https://blog.newrelic.com/product-news/new-relic-edge-ga/).

**5. What’s new**

When there are product updates, New Relic One notifies you and directs you to “What’s new”—your in-product destination to learn more about what we’ve released. There, you will find posts for each new feature with a brief description, resources, tips-and-tricks, and best practices to ensure you are productively leveraging the latest innovations and getting the most out of New Relic One. Learn more [here](https://blog.newrelic.com/product-news/whats-new-new-relic-one/).

**6. Anomalies visible in the activity stream**

You can now view anomalies with the activity stream, which is visible from the New Relic One homepage, APM summary page, and APM list page. The activity stream displays recent events from alerts and deployments and provides a direct view into what has changed in your system so that you can fix outages quickly. Learn more [here](https://docs.newrelic.com/whats-new/anomalies-visible-activity-stream).

**7. Kafka Connect**

Unlock open source and alternative instrumentation sources by ingesting data from Apache Kafka topics into New Relic One, without writing a single line of code, with the New Relic connector for Kafka Connect. Learn more [here](https://docs.newrelic.com/whats-new/kafka-connect-unlock-open-source-alternative-instrumentation-sources).

**8. Windows support for logs**

You can now send all of your logs to New Relic using the infrastructure agent in Windows. This feature includes new filters to select exactly which log types (application, security, or system) and messages you are interested in, all the way down to the EventID. Learn more [here](https://docs.newrelic.com/docs/integrations/host-integrations/host-integrations-list/windows-event-log-integration).

**9. Share with permalink**

Share insights, dashboards, and curated views from anywhere in New Relic One by clicking the permalink icon to copy a short, permanent URL to your clipboard automatically. Learn more [here](https://docs.newrelic.com/whats-new/share-dashboards-curated-views-permalinks).

**10. Scheduled alert muting**

Now you can schedule when you want to mute alerts to avoid messages during maintenance or deployments. Learn more [here](https://docs.newrelic.com/docs/alerts-applied-intelligence/new-relic-alerts/alert-notifications/muting-rules-suppress-notifications).

**11. Bonus! Observability for Good**

At our [Nerd Days 1.0 event](https://developer.newrelic.com/nerd-days/), New Relic introduced a new partnership with [Code for America](https://www.codeforamerica.org/) and announced [The Observability for Good Program](https://blog.newrelic.com/product-news/introducing-observability-for-good-program/), a new product donation program designed to help NGOs, nonprofits, and charities worldwide leverage the benefits of observability to advance their missions. [Eligible organizations](https://docs.newrelic.com/docs/accounts/accounts/subscription-pricing/eligibility-guidelines-new-relic-nonprofit-program) get:

* **1TB** of free data ingest per month in the [Telemetry Data Platform](https://newrelic.com/platform/telemetry-data-platform)
* Up to **5 free standard users** with access to [Full-Stack Observability](https://newrelic.com/platform/full-stack-observability)
* **50% discount** on [Applied Intelligence](https://newrelic.com/platform/applied-intelligence)
* Additional platform discounts for use beyond the free tier
* Access to ticketed support services
* **Access to**[**NewRelic.org**](https://newrelic.org/)**programs**, such as our pro bono program, where New Relic employees will volunteer to help not-for-profit customers with scoped technical projects to ensure optimal use of New Relic One

**How to Troubleshoot Performance and Crash Issues Faster with New Relic Real-Time Profiling for Java**

*This post was updated on November 3, 2020. It originally ran on March 17, 2020.*

Troubleshooting performance bottlenecks in your Java application or service can help you better understand where you’re wasting resources with inefficient processing. Additionally, if an incident occurs, you want to know what happened during the incident and what performance issues led up to it. To make such troubleshooting faster and easier, you need to see the high-fidelity runtime characteristics of your code running on the JVM—and you need that data in real time.

To that end, New Relic is excited to announce the availability of real-time Java profiling and the JVM cluster timeline view. Because of its low overhead, you can use real-time Java profiling in production environments to run continuous profiling of your Java code. The accompanying JVM cluster timeline view provides a fast and intuitive way to diagnose cluster-wide performance problems; for example, you can now quickly see how an application’s deployment affects the overall health of the cluster.

**Thanks to the Java community**

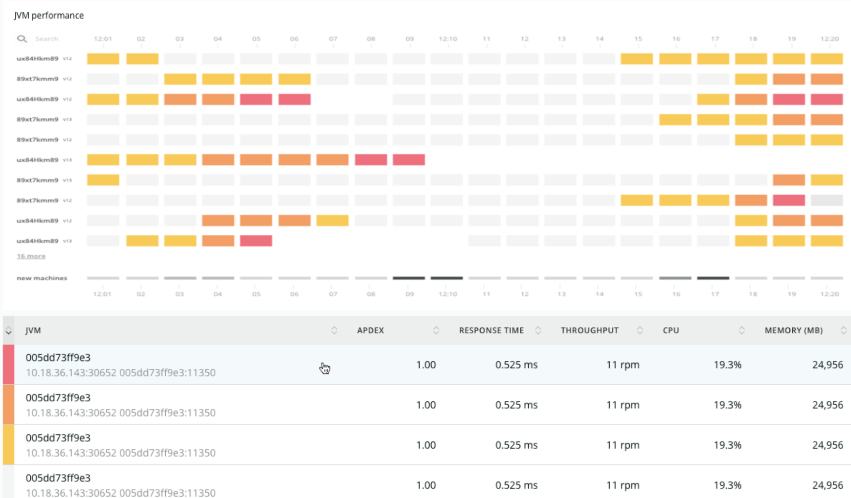
The profiling data New Relic uses for real-time Java profiling comes from the contributions of the Java community. After the release of JDK 9, Oracle changed the release model of Java and open-sourced the [Java Flight Recorder (JFR)](https://docs.oracle.com/javacomponents/jmc-5-4/jfr-runtime-guide/about.htm#JFRUH170). (To learn more about JFR and JFR Event Streaming, [check out this article](https://blogs.oracle.com/javamagazine/java-flight-recorder-and-jfr-event-streaming-in-java-14?source=:em:nw:mt::RC_WWMK190726P00001:NSL400036067&elq_mid=156196&sh=2419091808192613082213293109241520&cmid=WWMK190726P00001C0006#anchor_2) from our own [Ben Evans](https://blog.newrelic.com/author/benevans/).)

**Understand JVM cluster behavior over time using the cluster timeline view**

We’ve consistently heard from our [customers](https://newrelic.com/resources/case-studies/Australia-Post) about the need to view historic profiles across the JVM cluster supporting a service or application. So we built a unified dashboard that helps you get immediate visibility for all your JVMs to understand cluster behavior over time. This enables quicker troubleshooting and issue detection; for example, at a glance you can:

* See when a JVM was shut down or restarted
* See how instances were affected by their noisy neighbors
* See how a recent deployment affected the rest of the JVM cluster
* Go back up to 24 hours to view the root cause of incidents

Each row of the timeline represents a specific JVM over time. Inside each row, a box represents a 5-minute period of that JVM’s life. Yellow, orange, and red traffic lights indicate anomalous behavior for a JVM, so you can drill down into that instance and the right time period when investigating errors or other performance issues.

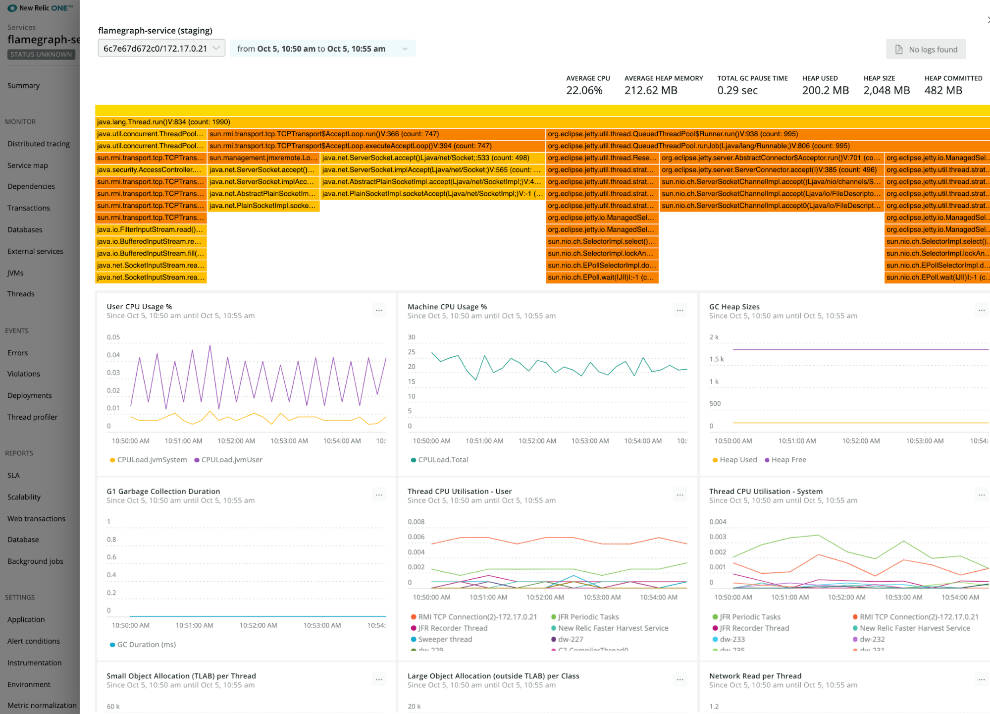


Speed troubleshooting with the JVM cluster timeline view

**Use the details panel to get insights into your JVM**

**1. Debug code execution performance using flamegraphs**

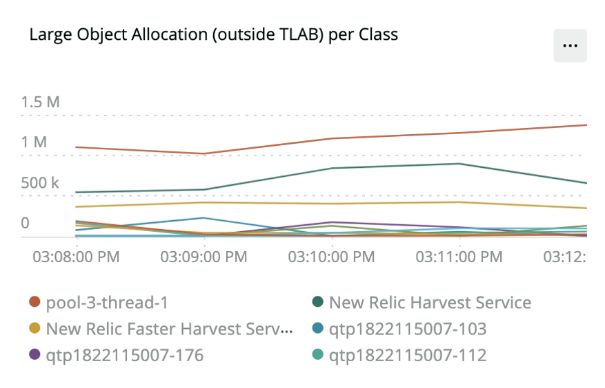
When debugging performance issues, you may discover that the bottleneck is in the running code. Use the **flamegraphs** view to find out where your application code is spending most of the execution time. This data is otherwise not available through logging or code instrumentation. You can use this to directly work on optimizing the hot spots in your code.



Diagnose bottlenecks in code performance with flamegraphs

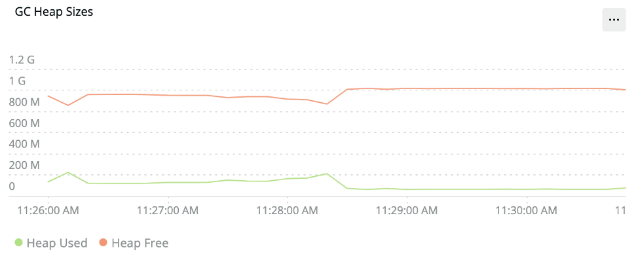
**2. How resources are allocated within a process**

In the detailed view of a JVM, the per-thread allocation buffer (TLAB) allocations graph shows you which threads are allocating the most resources. From this graph, you can see the individual events where new allocation buffers are handed out to application threads, which provides a much more accurate view of resources within a process.

[](https://blog.newrelic.com/wp-content/uploads/java_profile2.png)

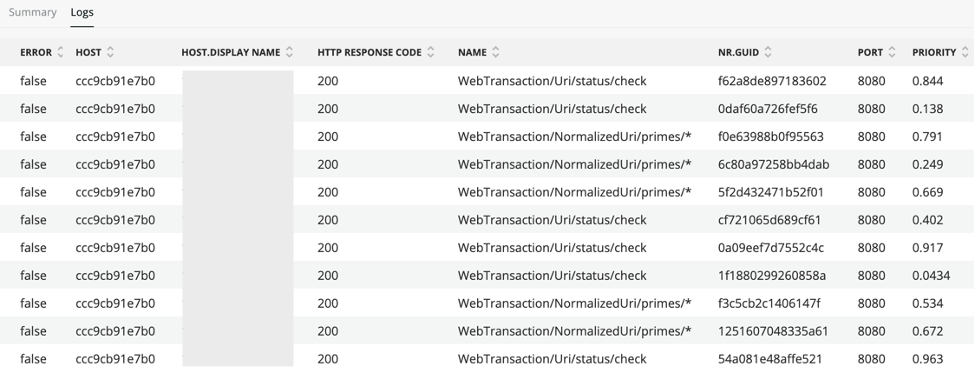
**3. Find inefficient garbage collection**

The main performance problems with garbage collections (GCs) are usually either that individual GCs take too long, or that too much time is spent in paused GCs (total GC pauses). The garbage collection (GC) graph shows garbage collection events over the lifetime of a JVM. The longest pause indicates where long garbage collection events have occurred over the selected time period while the overall time shows the total time spent in GC within a given time period.

[](https://blog.newrelic.com/wp-content/uploads/java_profile3.png)

**4. Connect profiling data with Logs**

If you’ve enabled [New Relic Logs](https://docs.newrelic.com/docs/logs/new-relic-logs/get-started/introduction-new-relic-logs), you can view logs for your Java application with data from the garbage collection graph to find transactions caught up in long garbage collection pauses.

[](https://blog.newrelic.com/wp-content/uploads/java_profile4a.png)

**Get started with real-time profiling for Java in New Relic One**

Real-time profiling for Java and the cluster timeline view are available in New Relic One, where you can incorporate this real-time profiling with other critical observability data.

To learn more about requirements and how to get started, [check out the documentation](https://docs.newrelic.com/docs/real-time-java-profiling-using-jfr-metrics). We work in the open, you can view the [source code](https://github.com/newrelic/newrelic-jfr-core#jfr-daemon) in GitHub. If you have any feedback, [let us know](https://discuss.newrelic.com/t/product-announcement-real-time-java-profiling/97199).

***Want to learn more about what’s happening in the Java ecosystem? Don’t miss***[***The State of Java: Trends And Data For One of the World’s Most Popular Programming***](https://blog.newrelic.com/technology/state-of-java/)