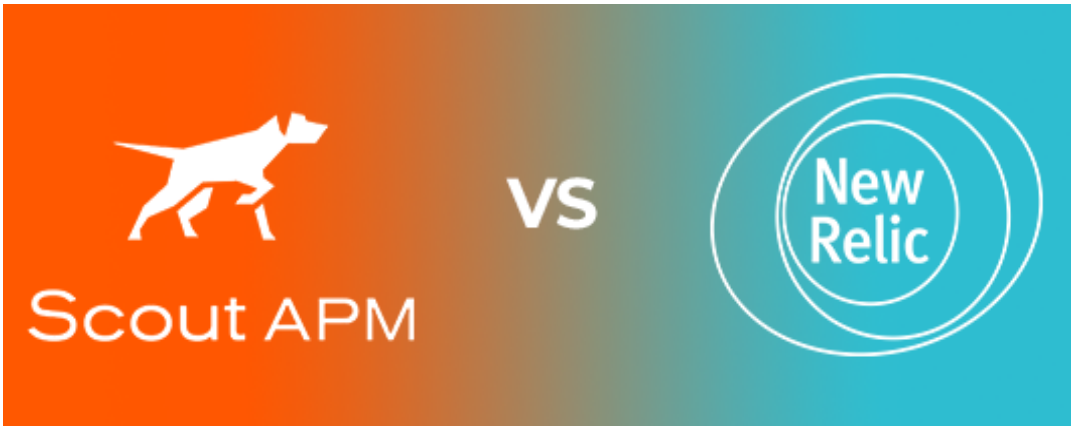
# Scout APM

## New Relic Comparison Guide

# Table of Contents




# Scout vs New Relic

There's no type of monitoring tool that can get to the heart of a performance problem faster than Application Performance Management (APM).

If you're making this decision, you may be choosing between New Relic and Scout. There are cases where New Relic is a better fit, clear cases where Scout fits like a glove, and some murky situations as well.

Where do you fall?

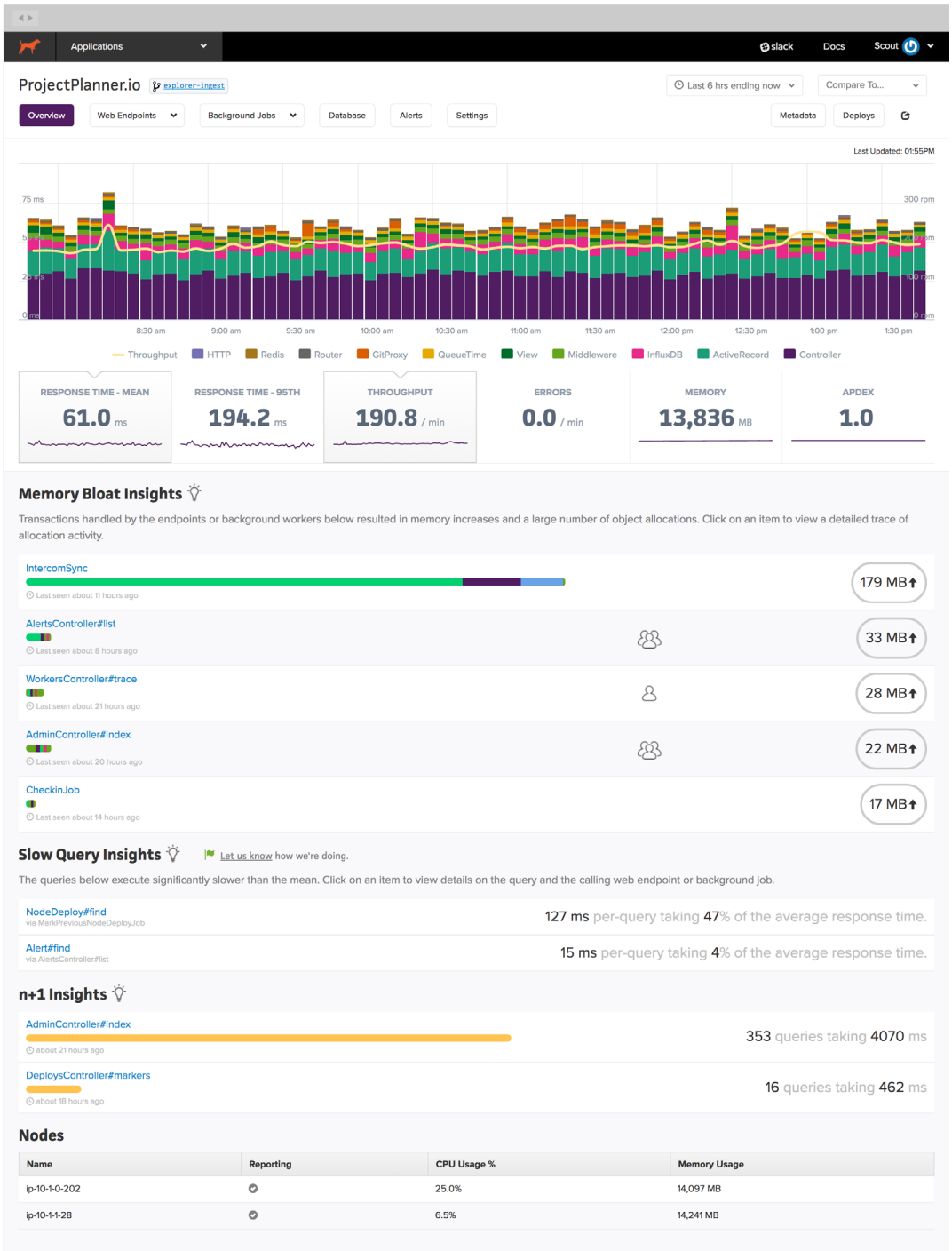
In this guide, we will summarize the factors to consider when choosing your best APM product.

## CHAPTER 1

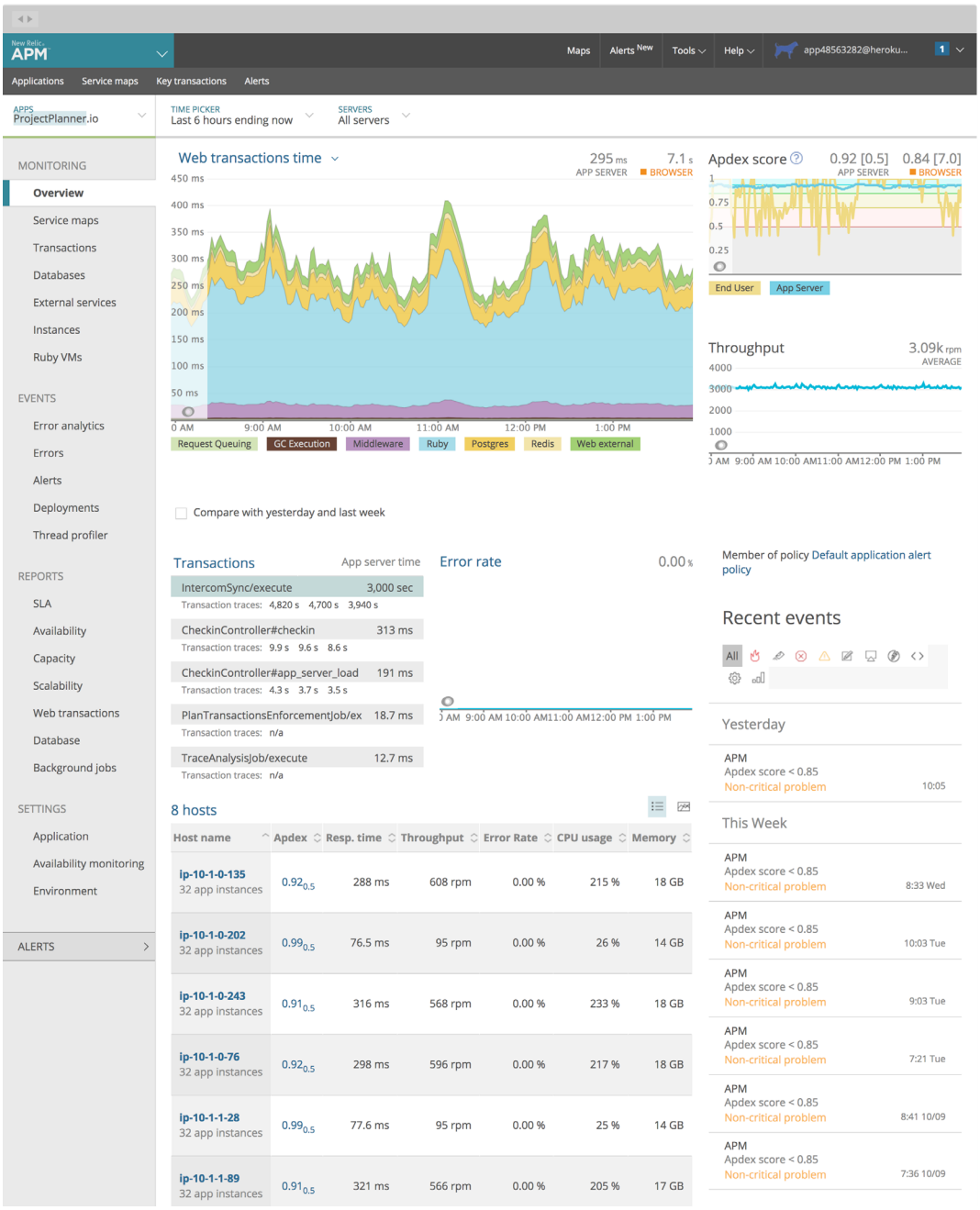
# App Overview

This is the first page you see when checking the health of your app. Scout's version is below and New Relic's on the following page:

## Scout App Overview



## New Relic App Overview




The top portion of the page is similar between New Relic and Scout: a breakdown of time spent by category (ex: Ruby, Database, External HTTP services, etc) over time. You can view data across similar timeframes in both Scout and New Relic (New Relic offers three months of data in their Pro package and Scout can do the same in their custom plans).

Both let you compare data to the past. Scout gives a few more options than New Relic, which is restricted to yesterday and last week comparisons.

The biggest difference with Scout's approach: several algorithms dig through your data and generate insights on performance directly on this page. For example, Scout identifies slow database queries, N+1 database queries, and memory bloat. These are ordered by impact, which help you decide where to get most improvements in the least amount of time.

Scout short-circuits the work a Site Reliability Engineer (SRE) often performs when investigating the health of an app.

Let's move from the 10,000 foot view to the soul of application monitoring.

## CHAPTER 2

# Language Support & Pricing

**Scout supports** Ruby, Elixir, PHP, Node.js, and Python with a Core Agent API that can be used to instrument any language. **New Relic supports** Ruby, Java, Node.js, PHP, .NET, Python, and Go.

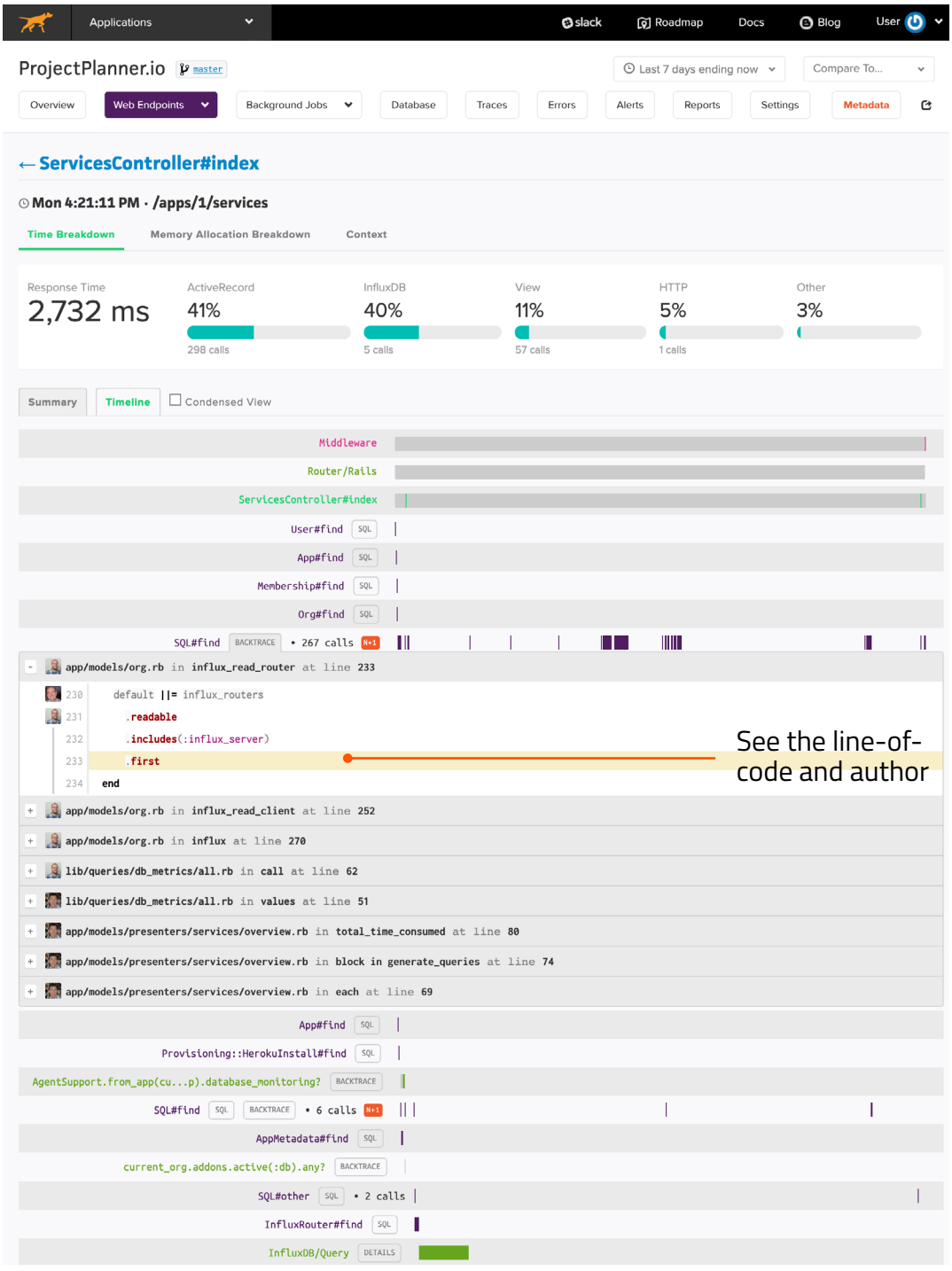
**Scout's monthly fee is about 30% lower than the monthly fee in a New Relic** annual contract. In most cases, Scout will also buy your New Relic contract.

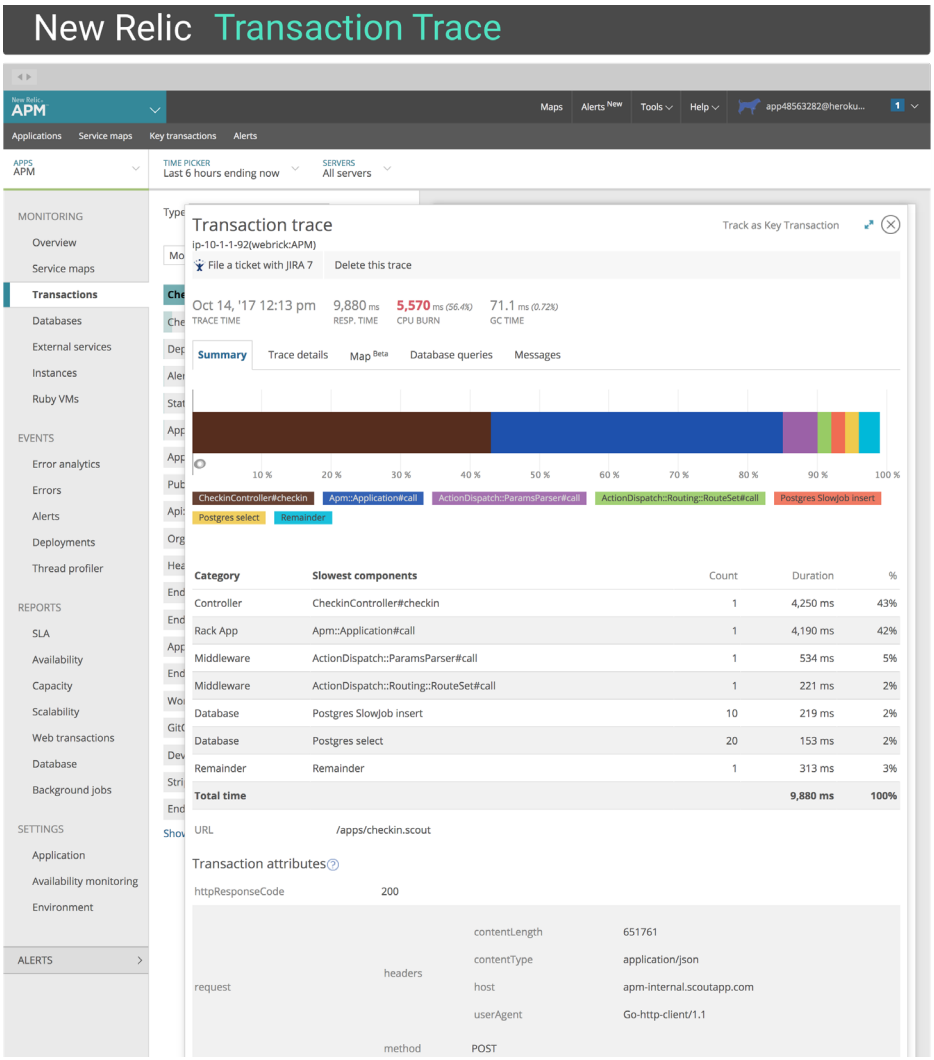
## CHAPTER 3

# Transaction Traces

The soul of app monitoring is the transaction trace: all the metrics and analysis an APM platform delivers originates from data collected in a trace. A transaction trace is a breakdown of time spent in a single web request or execution of a single background job.

A Scout trace is below and New Relic is on the next page:






There are some differences here as well:

- Scout provides a breakdown of memory allocations in addition to timing metrics. New Relic does not.
- New Relic offers cross-application tracing if your app touches another app you've monitored with New Relic. Scout does not.
- Scout is more likely to provide backtraces to slow method calls and can also display the code directly in the browser.
- Scout indicates the number of rows returned by SQL queries
- Scout can breakdown the time spent in custom code via ScoutProf. New Relic requires custom instrumentation.

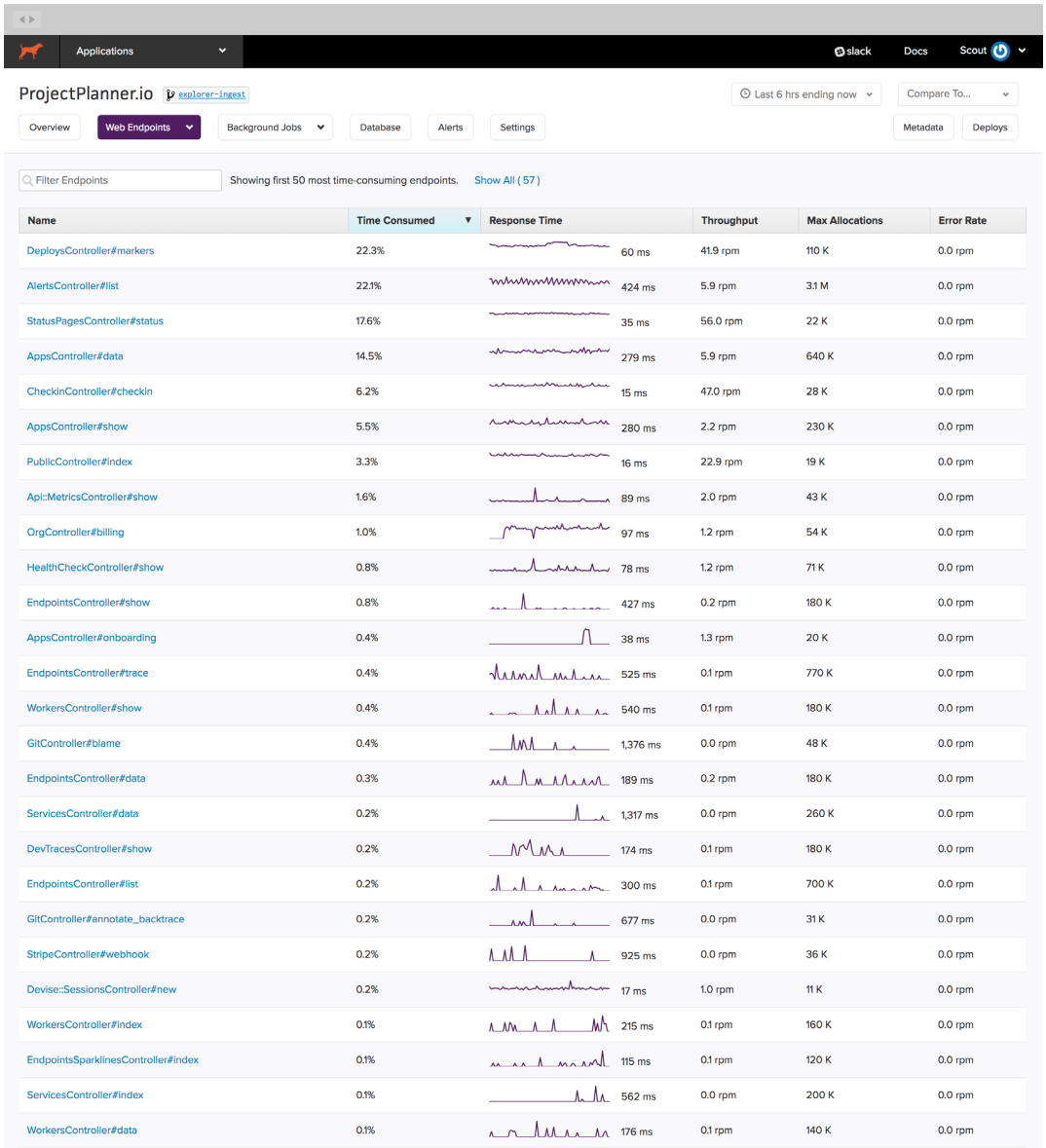
Scout typically collects more traces than New Relic. Both New Relic and Scout have similar, one-week retention periods for traces.

## CHAPTER 4

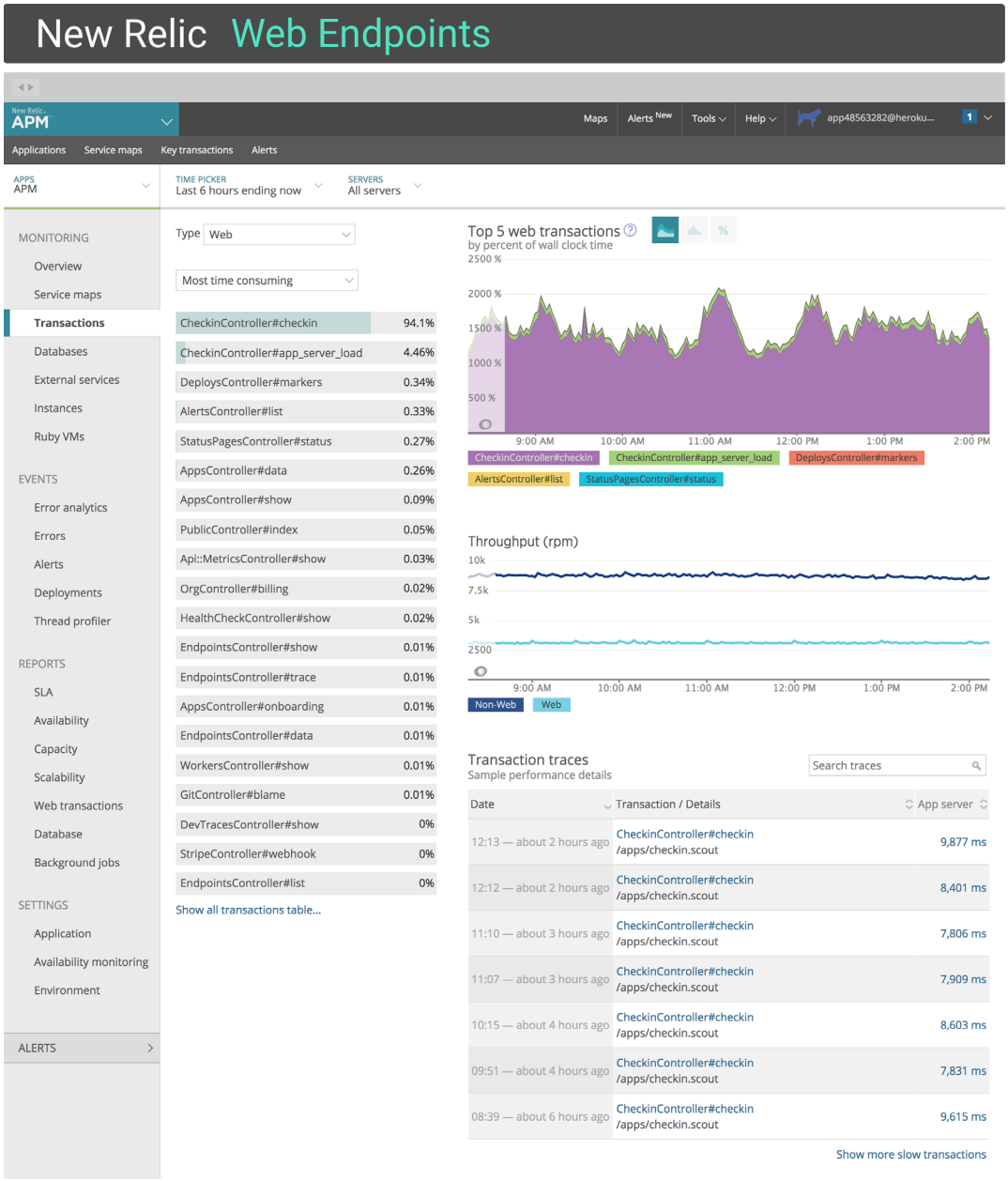
# Web Endpoints

Both New Relic and Scout let you view summary data on web endpoints in list format. Scout is listed first, New Relic second:

## Scout Web Endpoints




| Name | Time Consumed | Response Time | Throughput | Max Allocations | Error Rate |
|---|---|---|---|---|---|
| DeploysController#markers | 22.3% | 60 ms | 41.9 rpm | 110 K | 0.0 rpm |
| AlertsController#list | 22.1% | 424 ms | 5.9 rpm | 3.1 M | 0.0 rpm |
| StatusPagesController#status | 17.6% | 35 ms | 56.0 rpm | 22 K | 0.0 rpm |
| AppsController#data | 14.5% | 279 ms | 5.9 rpm | 640 K | 0.0 rpm |
| CheckinController#checkin | 6.2% | 15 ms | 47.0 rpm | 28 K | 0.0 rpm |
| AppsController#show | 5.5% | 280 ms | 2.2 rpm | 230 K | 0.0 rpm |
| PublicController#index | 3.3% | 16 ms | 22.9 rpm | 19 K | 0.0 rpm |
| Api::MetricsController#show | 1.6% | 89 ms | 2.0 rpm | 43 K | 0.0 rpm |
| OrgController#billing | 1.0% | 97 ms | 1.2 rpm | 54 K | 0.0 rpm |
| HealthCheckController#show | 0.8% | 78 ms | 1.2 rpm | 71 K | 0.0 rpm |
| EndpointsController#show | 0.8% | 427 ms | 0.2 rpm | 180 K | 0.0 rpm |
| AppsController#onboarding | 0.4% | 38 ms | 1.3 rpm | 20 K | 0.0 rpm |
| EndpointsController#trace | 0.4% | 525 ms | 0.1 rpm | 770 K | 0.0 rpm |
| WorkersController#show | 0.4% | 540 ms | 0.1 rpm | 180 K | 0.0 rpm |
| GitController#blame | 0.4% | 1,376 ms | 0.0 rpm | 48 K | 0.0 rpm |
| EndpointsController#data | 0.3% | 189 ms | 0.2 rpm | 180 K | 0.0 rpm |
| ServicesController#data | 0.2% | 1,317 ms | 0.0 rpm | 260 K | 0.0 rpm |
| DevTracesController#show | 0.2% | 174 ms | 0.1 rpm | 180 K | 0.0 rpm |
| EndpointsController#list | 0.2% | 300 ms | 0.1 rpm | 700 K | 0.0 rpm |
| GitController#annotate_backtrace | 0.2% | 677 ms | 0.0 rpm | 31 K | 0.0 rpm |
| StripeController#webhook | 0.2% | 925 ms | 0.0 rpm | 36 K | 0.0 rpm |
| Devise::SessionsController#new | 0.2% | 17 ms | 1.0 rpm | 11 K | 0.0 rpm |
| WorkersController#index | 0.1% | 215 ms | 0.1 rpm | 160 K | 0.0 rpm |
| EndpointsSparklinesController#index | 0.1% | 115 ms | 0.1 rpm | 120 K | 0.0 rpm |
| ServicesController#index | 0.1% | 562 ms | 0.0 rpm | 200 K | 0.0 rpm |
| WorkersController#data | 0.1% | 176 ms | 0.1 rpm | 140 K | 0.0 rpm |



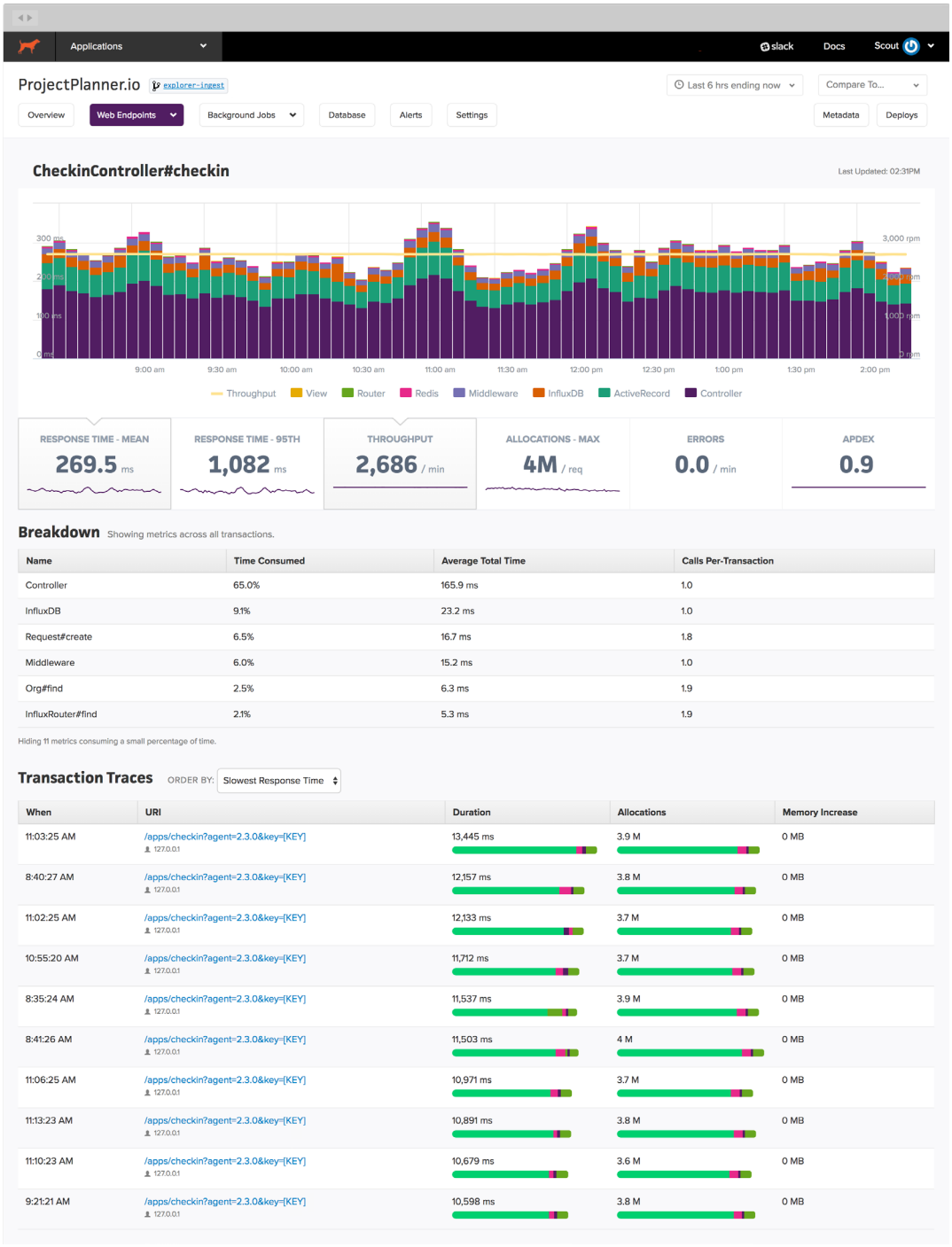

New Relic opts for several different visualizations on this page. Scout goes with a filterable, sortable view.

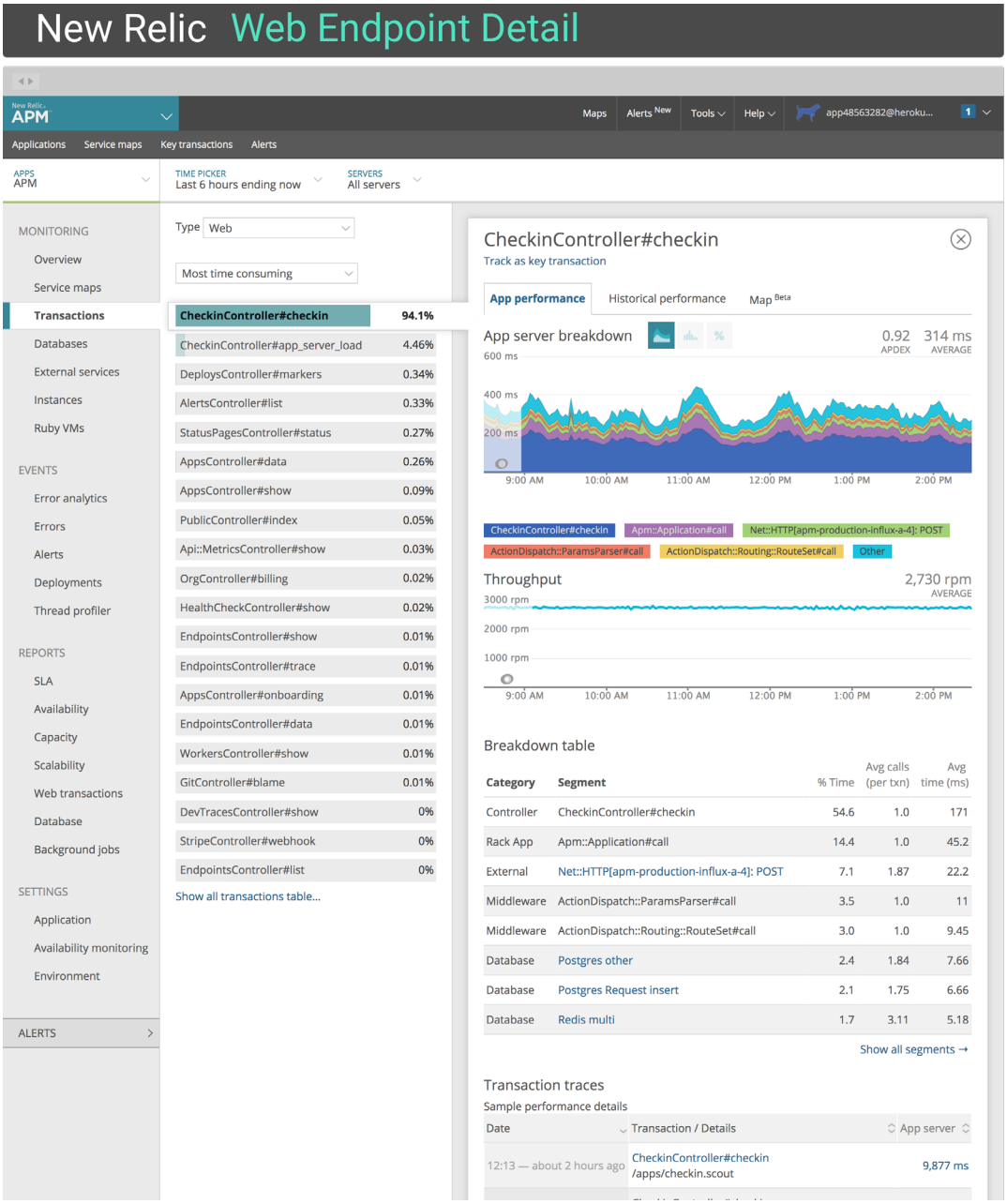
Both sort web endpoints by percent time consumed by default, helping you focus on the controller-actions that consume the greatest amount of time.

# CHAPTER 5
# Web Endpoint Detail

In the same way the app overview provides a breakdown of time spent, both New Relic and Scout offer endpoint-specific views for this:

## Scout Web Endpoint Detail





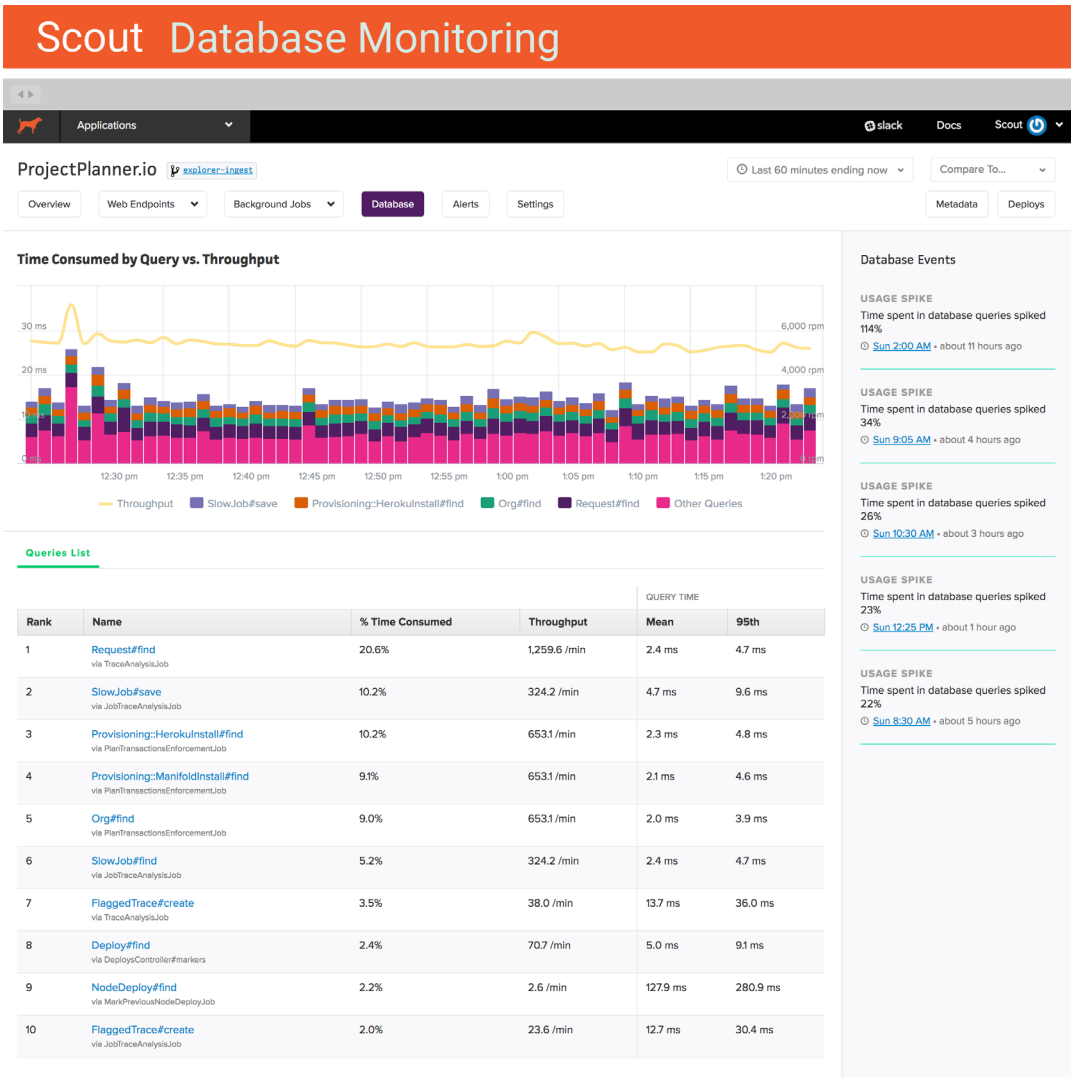

There are some differences on this page:

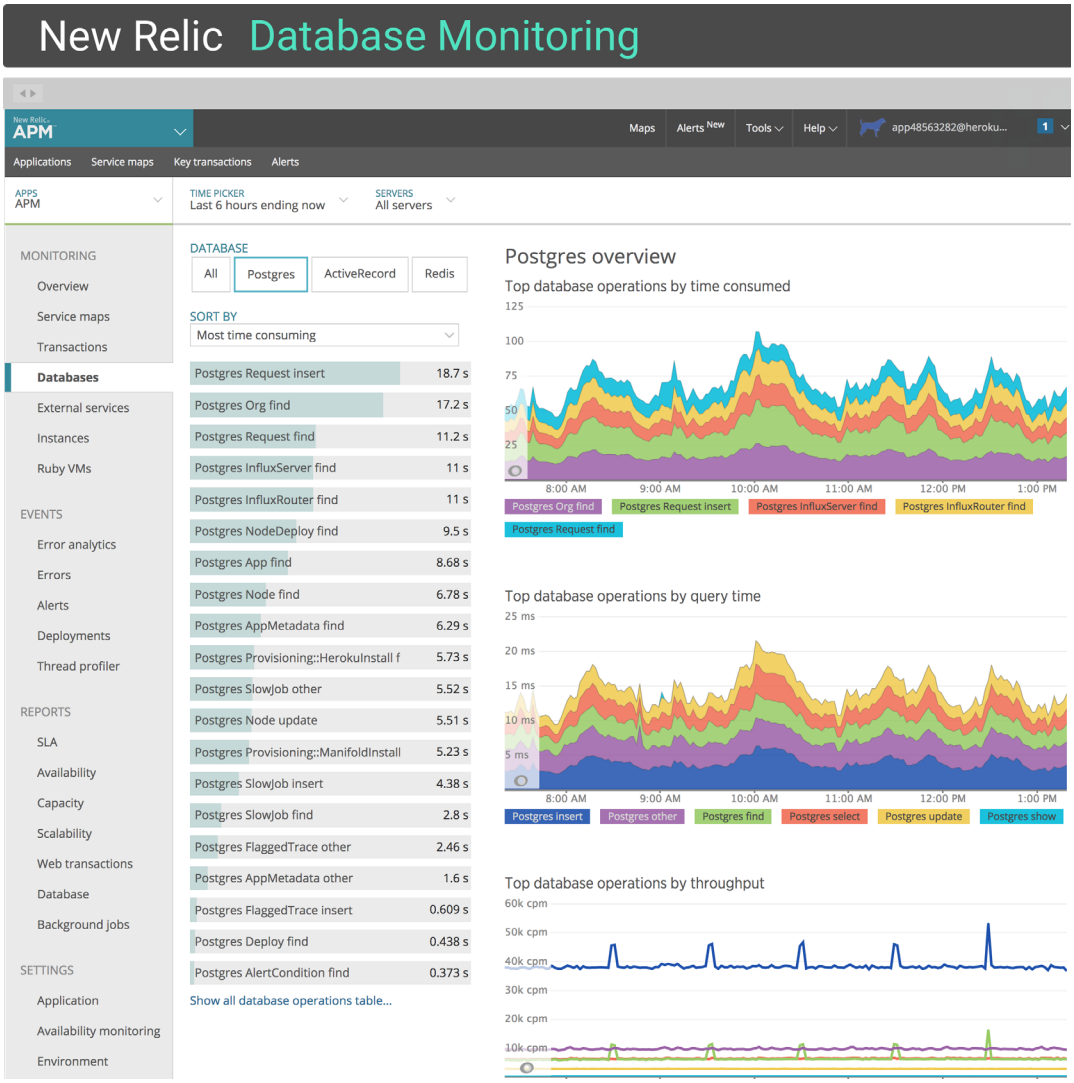
- New Relic offers a histogram display option of response times and Scout does not.
- Scout offers memory allocation metrics (more allocations lead to increased memory usage) while New Relic does not.
- New Relic's breakdown provides more categories, while Scout focuses the breakdown on database queries.
- Scout typically provides more transaction traces (up to 10 per-minute) and more options for sorting and selecting traces.

CHAPTER 6

# Database Monitoring

The database is the most common bottleneck for web applications. Since a database is a shared resource, one expensive query may cause many other types of queries to run slower. Both tools provide additional analytics around database query performance:





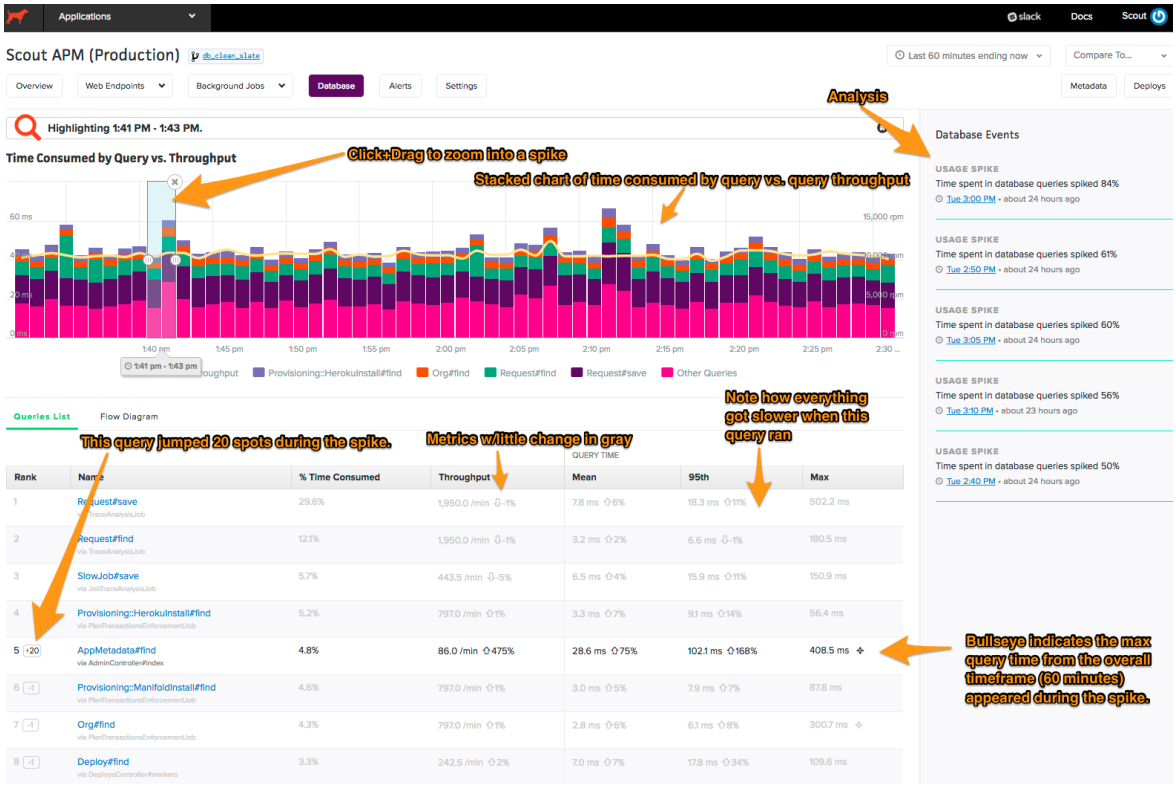

Both of the UIs resemble each tool's web endpoints display but made specifically for database queries. There are some subtle yet important differences between the two:

- Scout's chart displays data across every query - not just the top five most expensive queries - by combining less expensive queries into an "other queries" metric. This gives a more complete picture of performance.

- New Relic provides monitoring beyond ActiveRecord. Scout focuses on ActiveRecord.

- Scout's list of queries includes the calling controller-action or background job. Expensive queriers are frequently triggered by a few hotspots in the code: this makes it easier to identify those spots at a glance.

Finally, Scout makes it easy to compare performance in a small slice of time to the normal performance. This isn't possible with New Relic - you can change the entire timezone, but you can't compare between a small slice of time and a larger one. Here's how this looks within Scout:




## CHAPTER 7

# Background Jobs

Both New Relic and Scout monitor the core Ruby background job frameworks: Sidekiq, Delayed Job, and Resque. The UI is similar to endpoints, so I haven't provided a screenshot here.

CHAPTER 8

# Development Profiler

When you are actively resolving a performance issue, it's helpful to get immediate feedback on the results as you code. Otherwise, you need to deploy and wait for fresh data to verify the change had the intended impact.

New Relic used to have a developer mode, but it has since been removed from the agent.

Scout DevTrace gives immediate access to traces in your local development environment by clicking on a speed badge in the lower left of your browser:

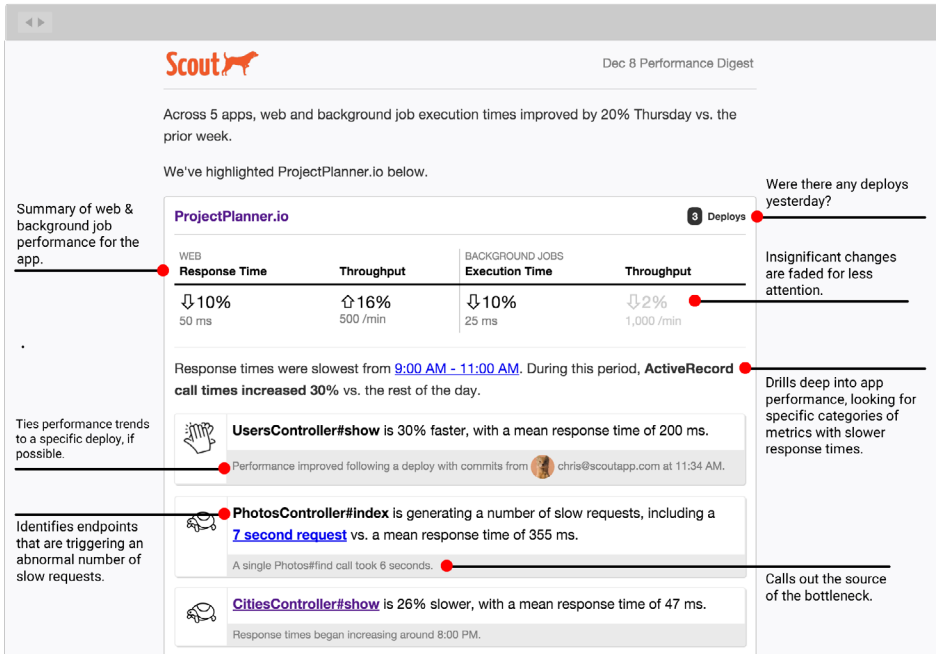

Slow feedback cycles - like a long-running test suite - are painful. DevTrace eliminates a lot of this as you can instantly verify that an optimization - like fixing an N+1 query with `includes` - is working as expected before deploying.

CHAPTER 9

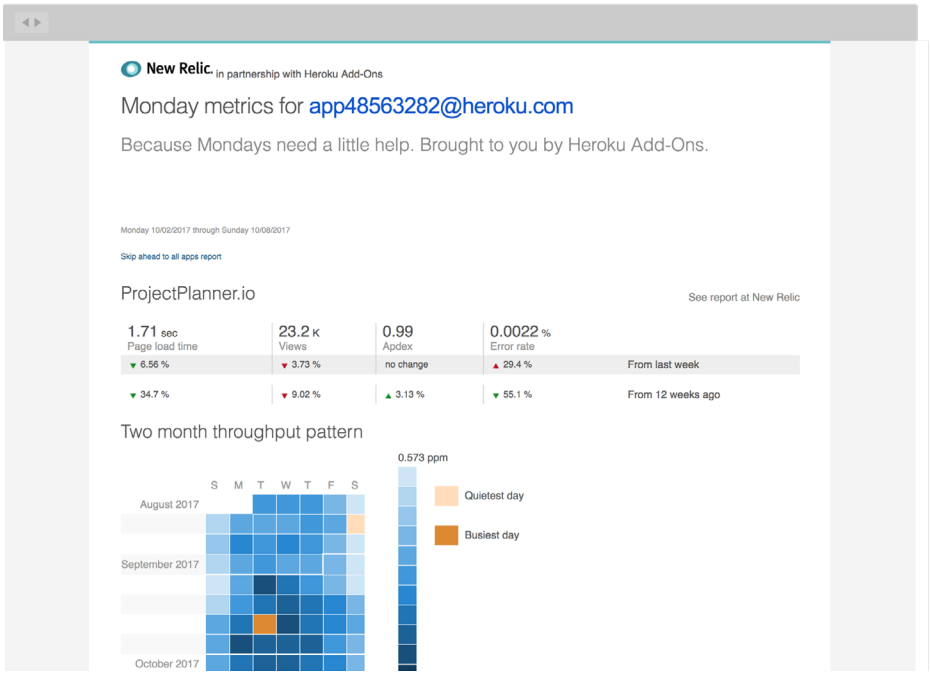
# Weekly Trends Email

A recurring report of your app's health is a great way for managers to stay in the loop on performance and to uncover slowly building problems. New Relic puts more emphasis on high-level numbers. Scout digs more into the source of problems and trends:

## Scout Weekly Email




## New Relic Weekly Email



## CHAPTER 10

# Agent, Alerting & Error Analytics

Both the New Relic and Scout agents are battle-tested. In Scout's open-sourced benchmarks, **Scout's agent was shown to have lower overhead**:

| APM Agent | Response Time (Mean) | Response Time (95th Percentile) | Response Time (Max) | Overhead |
|---|---|---|---|---|
| **None** | 55.6 ms | 106.4 ms | 2,174.1 ms | |
| **New Relic** | 80.4 ms | 149.5 ms | 2,263.5 ms | 44.5% |
| **Scout APM** | 56.8 ms | 102.7 ms | 2,168.7 ms | 2.2% |

The impact of agent overhead increases as CPU resources on your app servers becomes more scarce.

Scout and New Relic have comparable **alerting functionality**.

**Scout tracks error rates and integrates with third-party services** like Rollbar to provide details on exceptions. New Relic can provide this data, including backtraces to exceptions.

## CHAPTER 11

# Conclusion

New Relic and Scout require little configuration - outside adding a dependency - to try. You can run Scout and New Relic at the same time without causing conflicts.

Our suggestion? Try Scout's free 14-day trial alongside New Relic, then ask: "which tool helps your team solve performance issues faster?"