Alerts Best Practices

Building Alert Policies that don't keep you up at night

Kajan Moorthy
Senior Solutions Consultant

 New Relic.

Agenda

1	Housekeeping + Prerequisites
2	Key Concepts and Best Practices
3	Hands-on labs
4	Q&A

Prerequisites

You will need:

1. A glitch account (free) www.glitch.com from the previous sessions

   .. Or

   Your own application/service monitored by New Relic

2. A New Relic account www.newrelic.com

   .. and License key (which you can find under the account settings drop down)

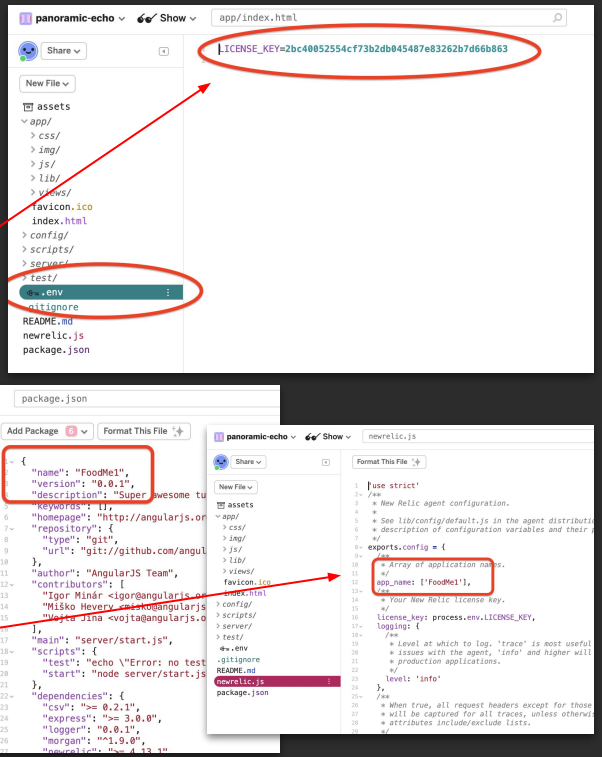
Let’s take 5 minutes to make sure you are ready to go.

5 mins

Lab: FoodMe Step 1

Clone FoodMe into a Glitch account

1. Create an account at glitch.com
2. Visit the url:
   https://glitch.com/edit/#!/remix/nru-foodme
   to clone foodme into your Glitch account
3. Edit the `.env` file to contain your New Relic license key
4. Set the name of the application in the Newrelic.js file and package.json file



Alerting - Best Practises

Getting started:
https://blog.newrelic.com/product-news/alerts-getting-started-best-practices/

Effective Alerting Strategy:
https://newrelic.com/resource/effective-alerting-guide

1. Make Alerts ‘actionable’
2. Alert as close to the source as possible
3. Test to start
   or use Baseline Metrics & adjust over time
4. Avoid too low thresholds (cause alert fatigue)
5. Disable alert conditions - eg. while testing others

5-7 mins

Lab: Create Alert Policies based on established thresholds

You have set up SLOs and SLA thresholds. Your CTO wants you to be proactive and add alerts that will notify you if the FoodMe application has degraded performance and is not meeting agreed levels (SLAs)

1. **Create a policy** to contain a series of conditions that monitor overall user experience.

   Use the following naming convention for the Policy:

   'service-name' User-Experience *'your-team-name'*

2. As it will have a range of conditions across the stack, you want to group any issues that are triggered together so that you can review and resolve them in one place if performance degrades.

   Which incident preference option should you choose?

3. Add the following conditions:

   - **APM > Backend Apdex drop below 0.85 for 5 minutes**. Add a warning of your choice
   - **Browser > Pageload time (2 seconds)** Add a warning of your choice

   Note: Use explicit condition names for the following conditions eg. 'service-name' 'condition' 'value exceeded'

5-7 mins

Lab: Alerting on a new application or unpredictable behaviour

The front end developers have started making use of AJAX calls to keep the client-server connections to a minimum and improve perceived performance.

The team is still experimenting with AJAX and want to keep doing so. There is no clear performance threshold yet but they still want to be notified when the AJAX response time deviates from the historical average.

1. Add an alert condition to your “User-Experience” policy you have set up already.

   Select the **Browser category > ‘Metric Baseline’** condition type.

   Choose **AJAX Response Time > Upper Only > for 3 minutes**

   - choose a sensitivity setting with the **slider**

NRQL Alerts

Query results

- Queries must return a number
- The alert condition works by evaluating that returned number against thresholds you set

Threshold Types

Static

- Condition based on the value returned

Baseline

- self-adjusting condition - based on the past behavior of the monitored values

Outlier

- Looks for values that are outliers from a FACET group

5-7 mins

Lab: Create Alert Policies using NRQL

Query your Custom Data and get alerted on it

You want to be prepared for busy patterns of sales on your FoodMe delivery site. You decide to use the NRQL Alert to trigger an incident when the average order value exceeds a set value.

1. Add an alert condition to the “User-Experience” policy you have set up already.

   Select the NRQL category, and use the following NRQL query:

```
SELECT average(orderTotal) AS 'Average
Order' FROM Transaction WHERE appName =
'my-FoodMe-app-name' AND name LIKE
'%api/order'
```

2. Use the Static Threshold type:

   - when query returns a value > above
   - 50 > at least once in > 3 minutes

   Add the condition name:

   - ‘Order Value Average > 50 FoodMe App’

   Set Violation time limit

   - 300 seconds

10 mins

Lab: Notify the right team

Use the Webhook Notification Channel

Now that the alert conditions have been configured, it's time to set up the notification channels.
For this specific Alert Policy, the Operations team needs to integrate the alerts with their Third Party Tool so you plan to use the **Webhook** channel option.

1. Choose an Operations `'teamName'` *(family-friendly !)*

2. Create a webhook notification channel that will send notifications to https://webhook.nru.to/
   - Alerts > Notification Channels > + New Notification
   - Channel Channel Type > Webhook
   - Channel Name: 'My *Operations team name*' Webhook
   - Base Url: https://webhook.nru.to/

3. Customise the webhook JSON Payload by adding an additional key value pair for your `'teamName'`:
   - `"teamName": "MyOpsTeamName-FoodMe",`
   - Save / Create Channel
   - Send a 'Notification Test' - Check for '200'
   - Check the Third Party Tool (indicator page!) https://alert-indicator.nru.to/

4. Start adding Food Delivery orders of over $50 value on your Glitch site
   - Does your indicator change colour??
   - Once it changes, Query your data and see what the average order value is. You can try out higher values...

10 mins

Homework Labs: Loss of Signal & AIOps

Using the new New Relic One Streaming Alerts and AIOps

Be notified if you stop receiving data from a source. Example: if your server went offline.

Proactively monitor SRE Golden Signals for your applications without defining alerts and thresholds.

Lab 1

1. Create a new Condition using NRQL.
2. Specify a NRQL query for Transactions for a Service and Add lost signal threshold.

Lab 2

1. Enable Proactive Detection.
2. Send notifications to your Slack channel.

Q&A

Thank You

learn.newrelic.com

New Relic.