# Alert Quality Management with Chaos Engineering




new relic

# Prerequisites

1. **Launch Instruqt when I say START.**

2. **Personal New Relic account.**

3. **Install Bernd's dashboard from:**

   https://github.com/berstr/aqm_workshop

# Prerequisites - QR Code

https://bit.ly/2024-01-o11y-day-stuttgart-aqm




new relic

# Top Five Common Mistakes

- Don’t know **what**, or the **right** thresholds
- Too many alerts, too many **false positives**
- No alerts maintenance or review process
- No business impact or end user in mind
- No proper description of alerts


**Alert First, Talk Later**

# Alert quality management

## Improving and optimizing the quality of your alerting

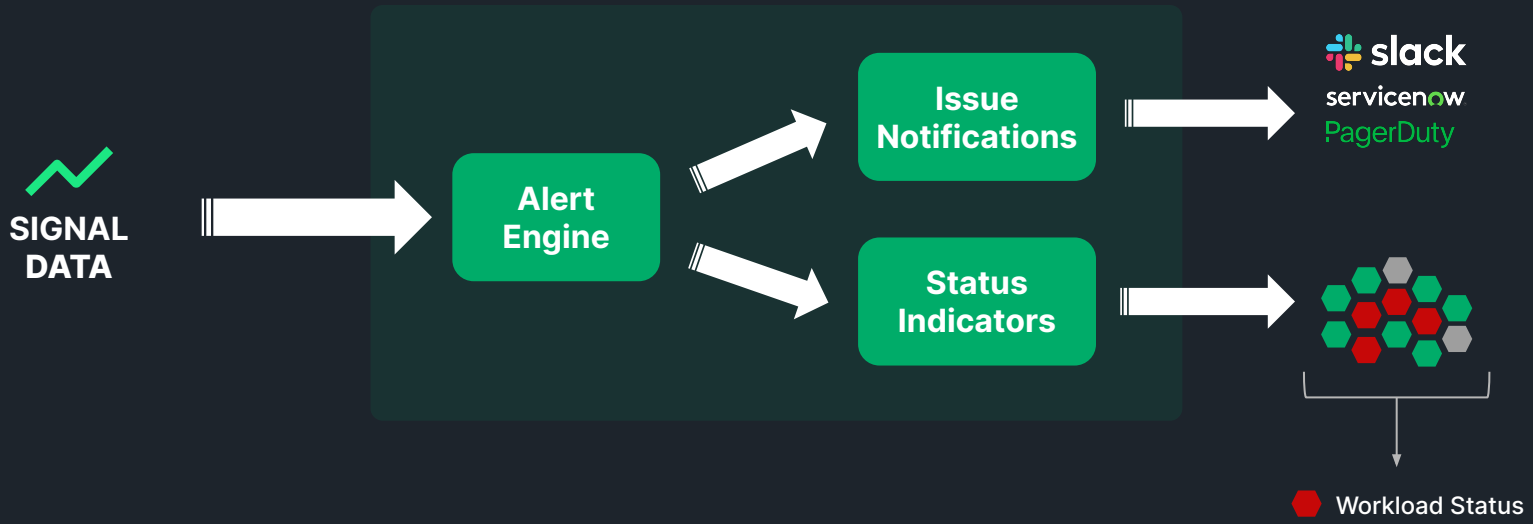
- Strategies to focus on **reducing the number of nuisance incidents** so that you focus only on alerts with **true business impact**
- Ensure that **fewer, more valuable**, incidents are created
- Includes **strategies, best practice** and **tooling** for measuring and analysing **alert quality**



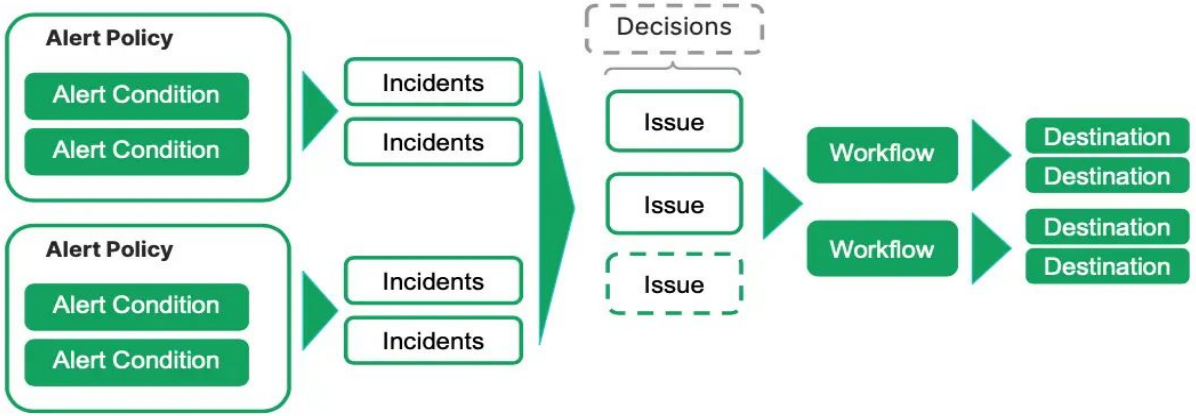
Documentation on alert quality management

new relic

# Getting Started with New Relic Alerts



# Getting Started with New Relic Alerts




**Policies** - groupings of conditions where you define how you get notified when a incident occurs.
**Conditions** - configuration objects that you set up to open an incident.
**Incidents** - events where a condition threshold is breached.
**Issues** - a collection of one or more incidents that cause a notification to be sent.
**Workflow** - control where you want to **receive** notifications about issues.
**Destination** - control where you want to **send** notifications about issues.

# 16 Challenges to Learn about AQM

## Alert Storm & Fatigue

- M.E.L.T
- NRQL - FACET

**Bonus!**
**Post Mortem**

## Improve Alert Response

- Error Inbox
- Naming
- Tags
- Anomaly Detection
- Runbook URL

## Improve Alert Quality

- Issue Preference
- Notification Method
- Enrichment
- Notification Integration
- Muting Alerts

## Improve Alert Maintenance

- Alerts Overview UI
- AQM Dashboard
- Mean Time to Close
- O11y as Code



new relic

# 16 Challenges to Learn about AQM

**Alert Storm & Fatigue**

- M.E.L.T
- NRQL - FACET

**Bonus!**
**Post Mortem**

**Improve Alert Response**

- Error Inbox
- Naming
- Tags
- Anomaly Detection
- Runbook URL

**Improve Alert Quality**

- Issue Preference
- Notification Method
- Enrichment
- Notification Integration
- Muting Alerts

**Improve Alert Maintenance**

- Alerts Overview UI
- AQM Dashboard
- Mean Time to Close
- O11y as Code

# What’s causing the spike in CPU?

## Challenge ONE

- Can you investigate what is causing the spike and where the script is running from?
- Without the use of **Events** and **Logs**, would it be difficult to identify the issue?
- What additional signals can we incorporate to improve **Mean Time to Recovery**?

# What's causing the spike in CPU?

## Challenge ONE

- Can you investigate what is causing the spike and where the script is running from?
- Without the use of **Events** and **Logs**, would it be difficult to identify the issue?
- What additional signals can we incorporate to improve **Mean Time to Recovery**?

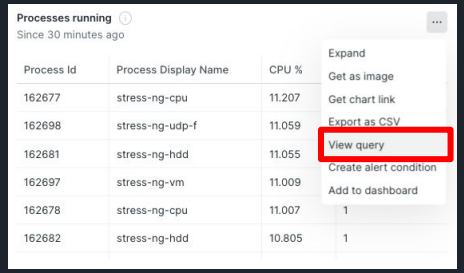
## Answer

**Processes - stress-ng**

**Script - /root/alertqualitymanagementWithNR/chaos/script1.sh**

# What's causing the spike in CPU?

## Challenge ONE




**Important!**
By default, the agent doesn't send data about the operating system's processes.
To enable the sending of process data set enable_process_metrics to true.
To fine-tune which processes you want to monitor, configure include_matching_metrics.
**View Query** is great to help you fine-tune your alerts.

```
-workshopaqm-infra CRON[242494]: (root) CMD (sh /root/alertqualitymanagementWithNR/chaos/script1.sh)
-workshopaqm-infra stress-ng: invoked with 'stress-ng --cpu 4 --vm 2 --vm-bytes 2G --hdd 4 --fork 8 --matrix 0 -
t 180s --udp-flood 0 -t 3m --timeout 3m --metrics' by user 0 'root'
```

**The location of the chaos script**

## Answer

**Processes - stress-ng**

**Script - /root/alertqualitymanagementWithNR/chaos/script1.sh**

# Experiencing Alert Storm & Fatigue

## Challenge TWO

- Can you identify which condition is using a special NRQL using **FACET**?
- **What will happen** when you turn on this special alert using FACET?

# Experiencing Alert Storm & Fatigue

## Challenge TWO

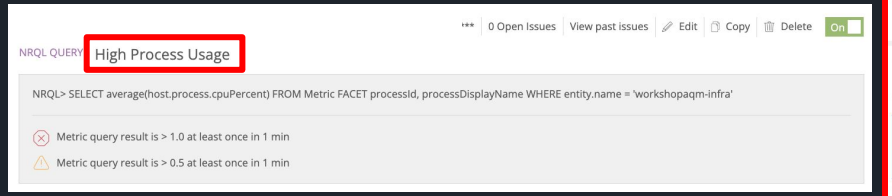
- Can you identify which condition is using a special NRQL using **FACET**?
- **What will happen** when you turn on this special alert using FACET?

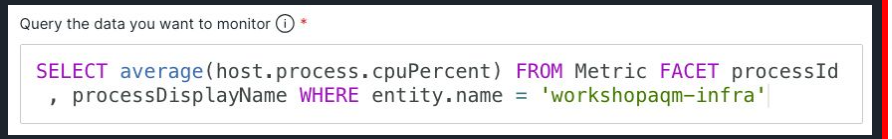
## Answer

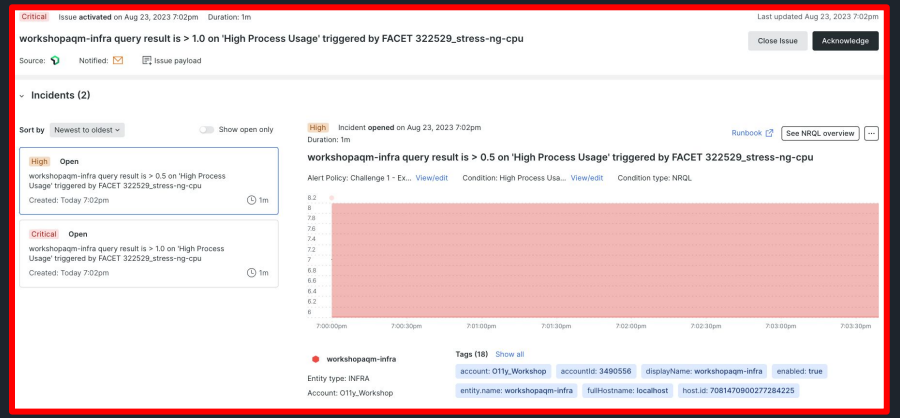
**High Process Usage**

# Experiencing Alert Storm & Fatigue

## Challenge TWO



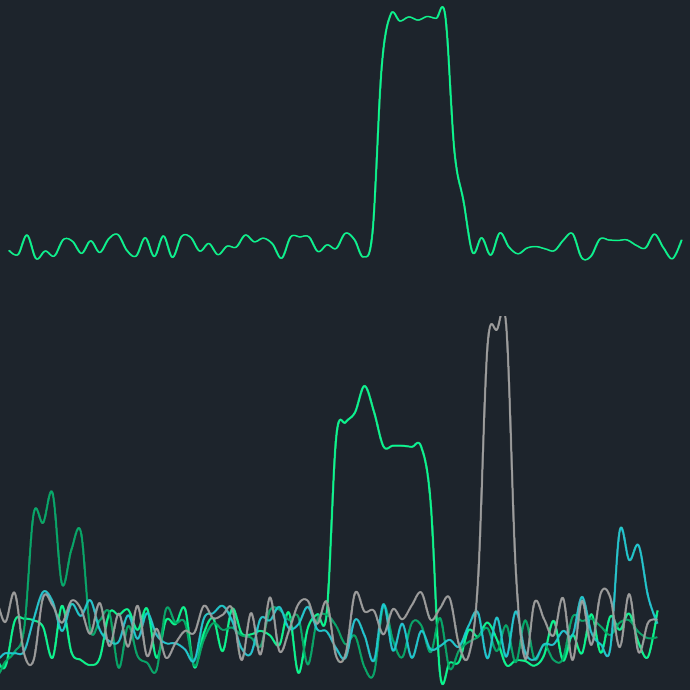

## Answer

### High Process Usage

Alert Storm & Fatigue

new relic

# What is a Signal?



## Signal

- **Any data** being sent to New Relic
- Result of an **NRQL query** against incoming data
- If you can **query it you can alert** on it!

## Faceted signal

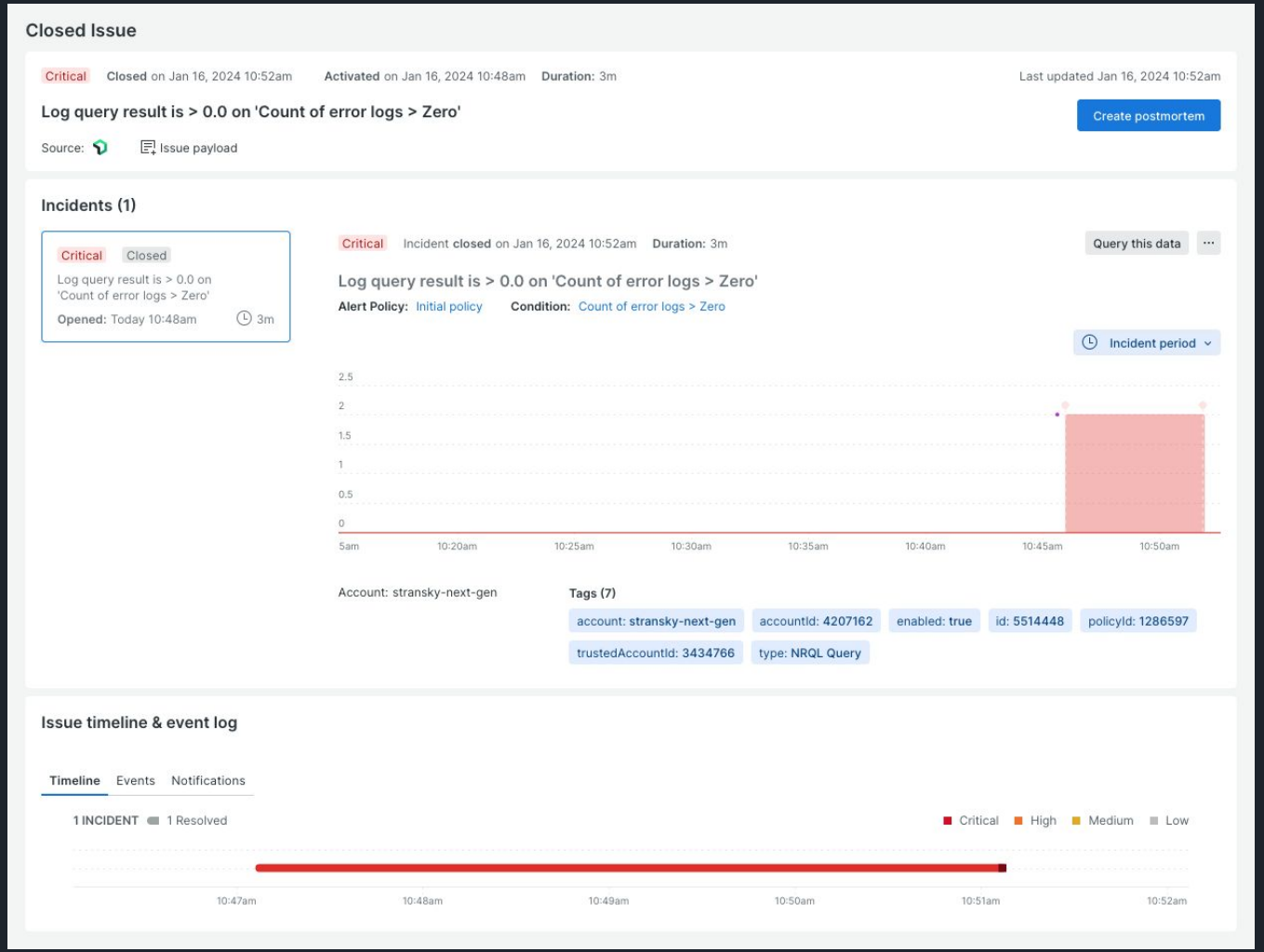
- Faceting the query results in **multiple signals**
- Each signal can trigger separate alerts
- Useful for large amounts of data points!

# Faceted signal

Alert Issue

Based on NRQL without FACET:

From Log Select count(*) where message lile ‘%error=%’

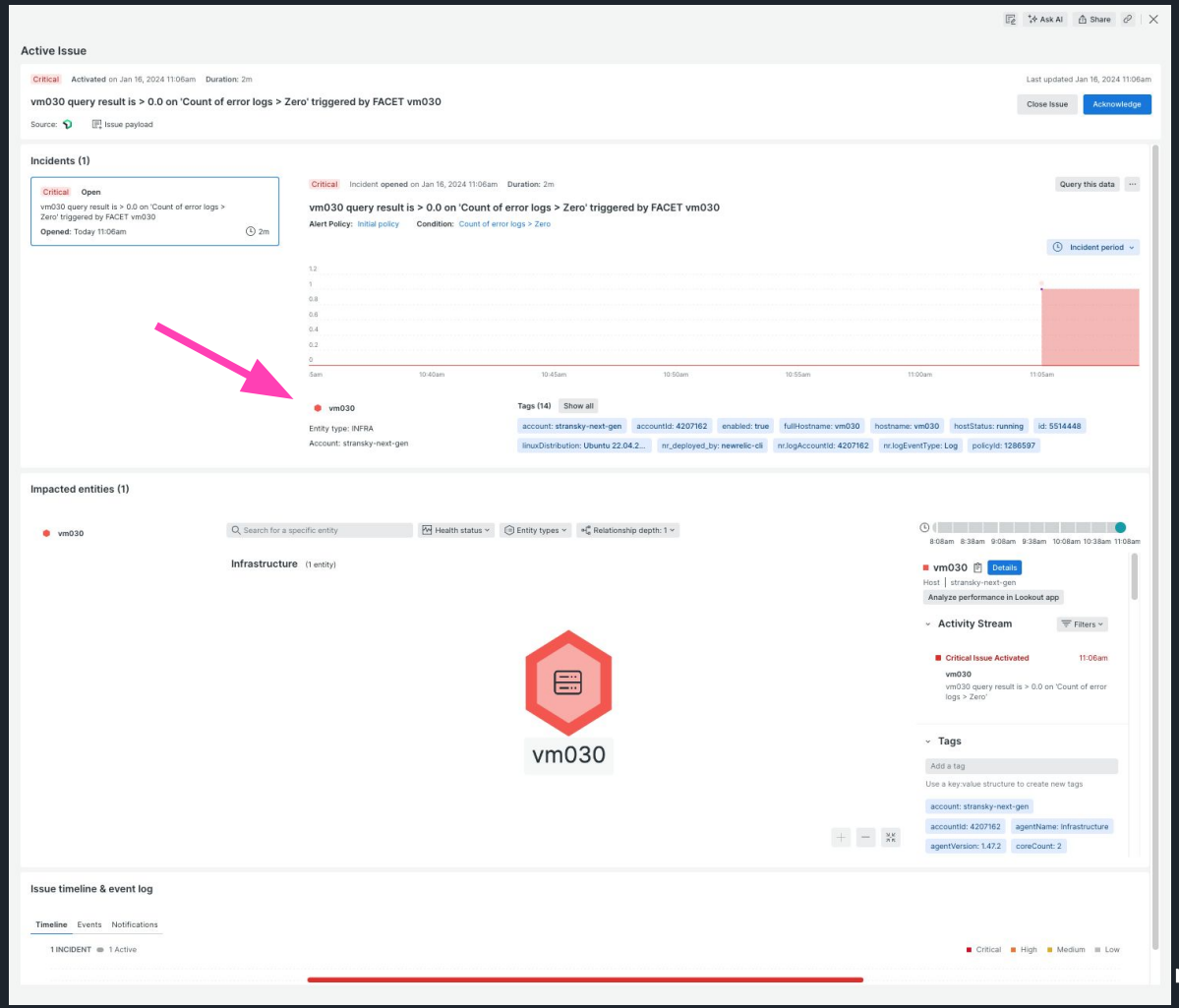


# Faceted signal

Alert Issue

Based on NRQL with FACET:

From Log Select count(*) where message lile ‘%error=%’ FACET hostname






relic

# Creating a Post Mortem

## Challenge BONUS

- Postmortem is a retrospective process that teams use to analyze what worked and what didn't when responding to and resolving an incident.
- In the platform, the postmortem feature is a tool you can use to prepare a successful retrospective process for your team.
- Postmortems **automatically collect data** related to an incident, freeing up your team to focus on analysis and action items for improved responses to future incidents.

# Creating a Post Mortem

## Challenge BONUS

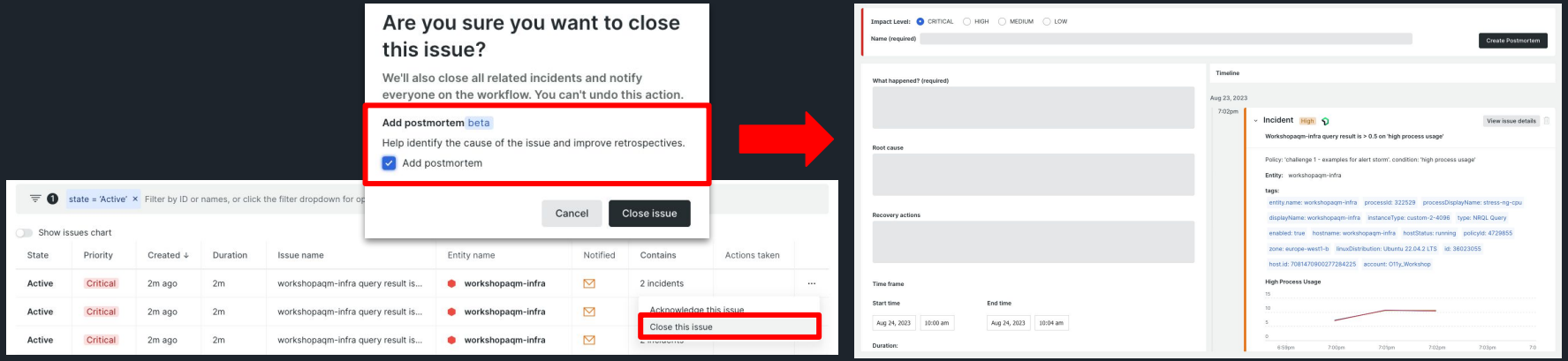
- Postmortem is a retrospective process that teams use to analyze what worked and what didn't when responding to and resolving an incident.
- In the platform, the postmortem feature is a tool you can use to prepare a successful retrospective process for your team.
- Postmortems **automatically collect data** related to an incident, freeing up your team to focus on analysis and action items for improved responses to future incidents.

## Answer

**From the issue feed, select an Open Issue for your postmortem.**

# Creating a Post Mortem

## Challenge BONUS




## Answer

**From the issue feed, select an Open Issue for your postmortem.**



new relic

# 16 Challenges to Learn about AQM

**Alert Storm & Fatigue**

- M.E.L.T
- NRQL - FACET

**Bonus!**
**Post Mortem**

**Improve Alert Response**

- Error Inbox
- Naming
- Tags
- Anomaly Detection
- Runbook URL

**Improve Alert Quality**

- Issue Preference
- Notification Method
- Enrichment
- Notification Integration
- Muting Alerts

**Improve Alert Maintenance**

- Alerts Overview UI
- AQM Dashboard
- Mean Time to Close
- O11y as Code

**Improve Alert Response**

new relic

# What's causing additional issues?

## Challenge THREE

- What went wrong in the game? **Can the game run successfully?**
- If we didn't add application-level insights, how would engineers resolve these issues?
- Why most engineers focus on the Infrastructure layer?
- Will players be able to enjoy the game without experiencing any major glitches or bugs?

**Improve Alert Response**

# What's causing additional issues?

## Challenge THREE

- What went wrong in the game? **Can the game run successfully?**
- If we didn't add application-level insights, how would engineers resolve these issues?
- Why most engineers focus on the Infrastructure layer?
- Will players be able to enjoy the game without experiencing any major glitches or bugs?

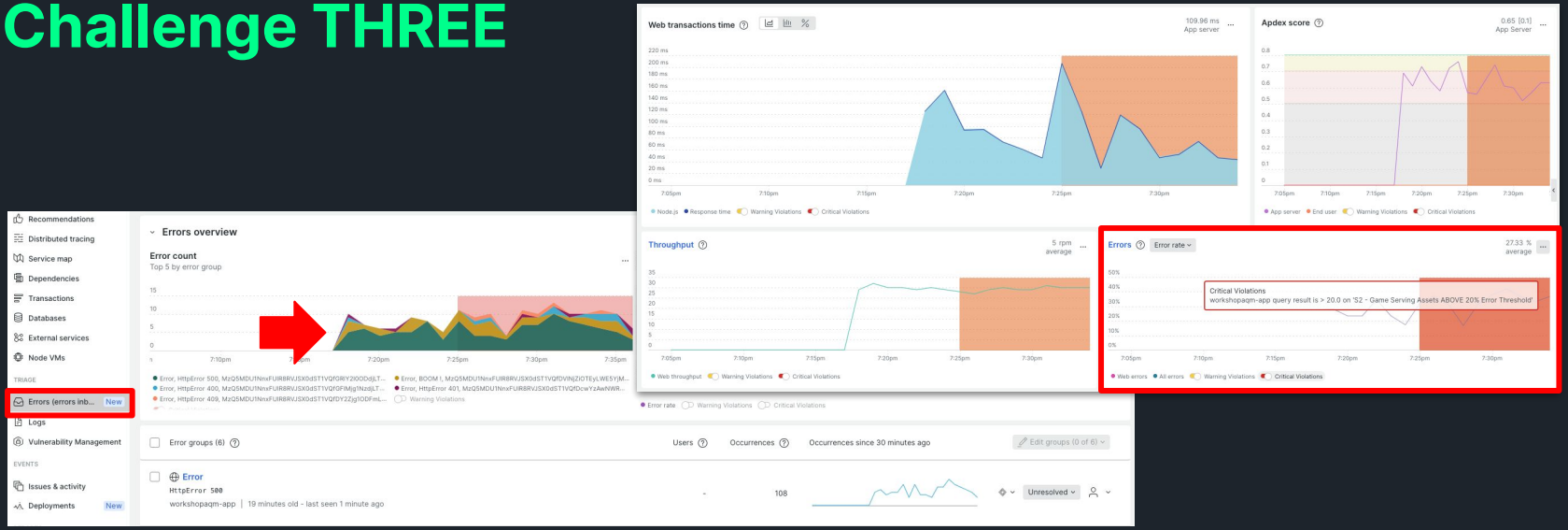
## Answer

**Errors (500, Boom, 401, 409) as seen in Error Inbox**

**Improve Alert Response**

# What's causing additional issues?

## Challenge THREE




## Answer

**Errors (500, Boom, 401, 409) as seen in Error Inbox**

**Improve Alert Response**

new relic

# Which alert contains S1?

## Challenge FOUR

- The description and the tags must give you **self-descriptive** alerts to know which service is wrong, which environment is involved, which team owns it, and if it's impacting to end users. It helps you to answer faster and decide what to do.
- Service categorization helps with **prioritization** and allocation of resources.

# Which alert contains S1?

## Challenge FOUR

- The description and the tags must give you **self-descriptive** alerts to know which service is wrong, which environment is involved, which team owns it, and if it's impacting to end users. It helps you to answer faster and decide what to do.
- Service categorization helps with **prioritization** and allocation of resources.

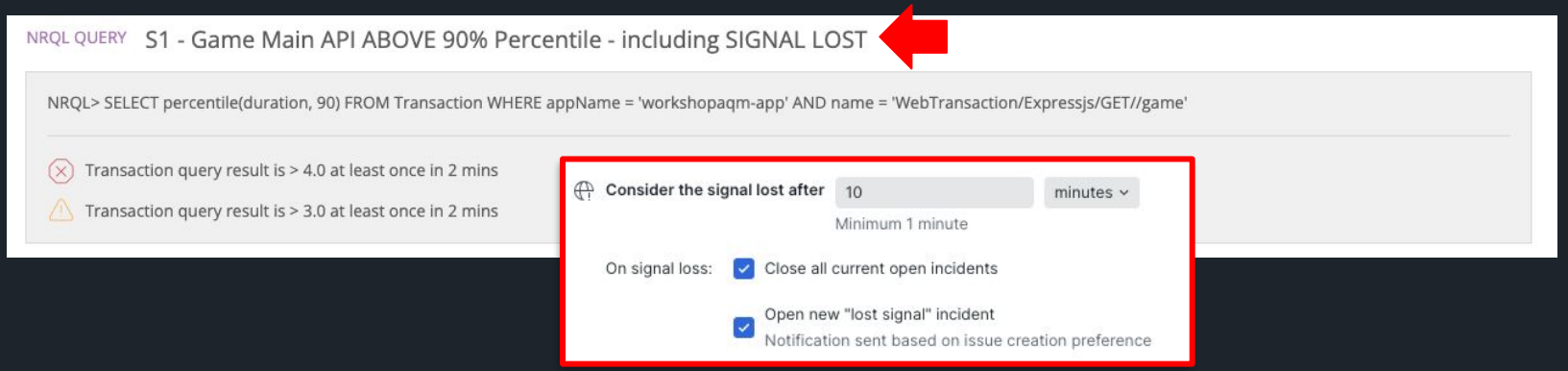
## Answer

**S1 - Game Main API ABOVE 90 Percentile - including SIGNAL LOST**

**Improve Alert Response**

# Which alert contains S1?

## Challenge FOUR




## Answer

**S1 - Game Main API ABOVE 90% Percentile - including SIGNAL LOST**

# Add tags to Terraform, or via the UI?

## Challenge FIVE

- Your issues and incidents have these tags in their metadata.
- Use them to do flexible filters in workflows or add them to your notification payload.
- Tags are extremely important for review, reporting, and analysis.

# Add tags to Terraform, or via the UI?

## Challenge FIVE

- Your issues and incidents have these tags in their metadata.
- Use them to do flexible filters in workflows or add them to your notification payload.
- Tags are extremely important for review, reporting, and analysis.

## Answer

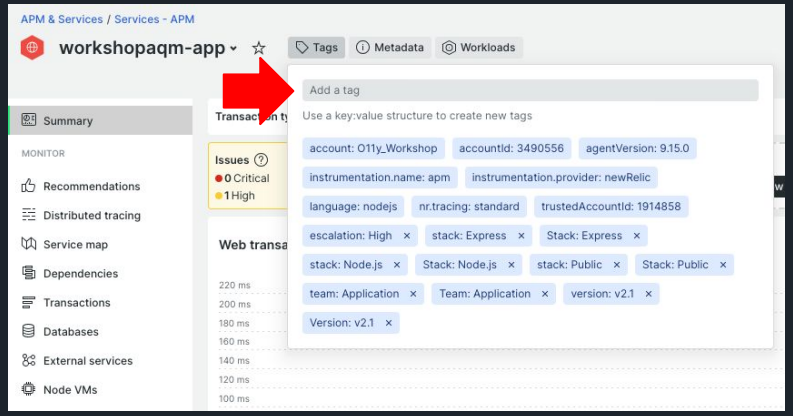
**resource newrelic_entity_tags in TF, or via New Relic Tags UI**

# Add tags to Terraform, or via the UI?

## Challenge FIVE

```
# resource to create, update, and delete tags for a New Relic entity - App
resource "newrelic_entity_tags" "appname" {
    guid = data.newrelic_entity.appname.guid

    tag {
        key = "escalation"
        values = ["High"]
    }

    tag {
        key = "stack"
        values = ["Node.js", "Express", "Public"]
    }

    tag {
        key = "version"
        values = ["v2.1"]
    }

    tag {
        key = "team"
        values = ["Application"]
    }
}
```

via Terraform




via the UI

## Answer

**resource newrelic_entity_tags in TF, or via New Relic Tags UI**

Improve Alert Response

new relic

# Configured with Anomaly Threshold?

## Challenge SIX

- Explore anomaly based alerts.
- Anomaly thresholds use past data to dynamically predict the data's near-future behavior. This will adjust over time as it learns the patterns of your data.
- Best for the environment when there is constantly available data.

# Configured with Anomaly Threshold?

## Challenge SIX

- Explore anomaly based alerts.
- Anomaly thresholds use past data to dynamically predict the data's near-future behavior. This will adjust over time as it learns the patterns of your data.
- Best for the environment when there is constantly available data.

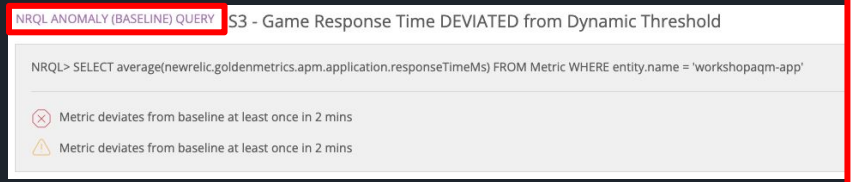
## Answer

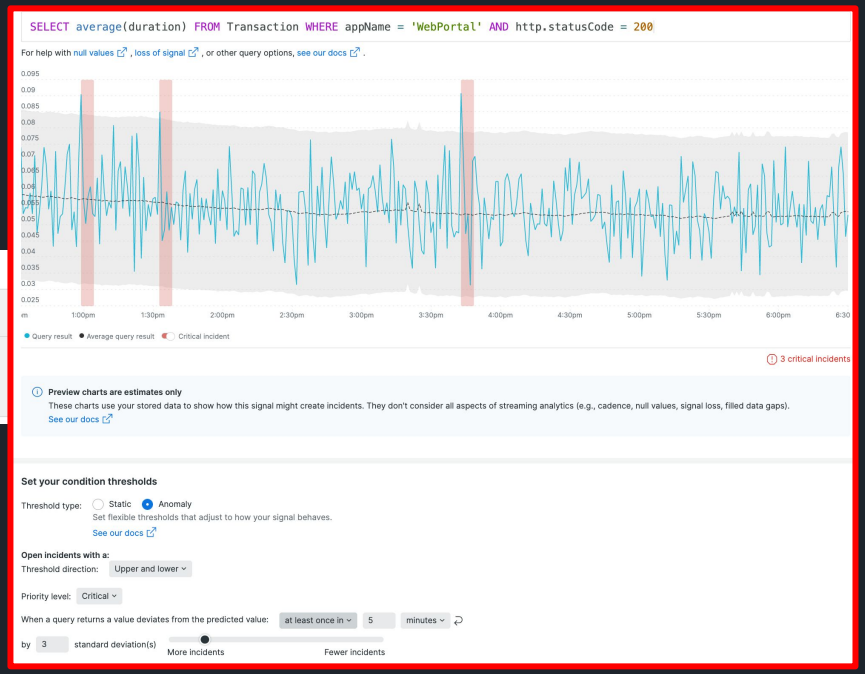
**All Alerts with S3**

**Improve Alert Response**

# Configured with Anomaly Threshold?

## Challenge SIX




## Answer

### All Alerts with S3

**Improve Alert Response**

new relic

# What is the current runbook URL?

## Challenge SEVEN

- The runbook must describe remediation steps to follow and who to involve or escalate to.
- Includes identifying the individuals or teams responsible for implementing each step and the channels that should be used to escalate the issue if necessary.
- Outline any preventive measures that can be taken to avoid similar incidents in the future.

# What is the current runbook URL?

## Challenge SEVEN

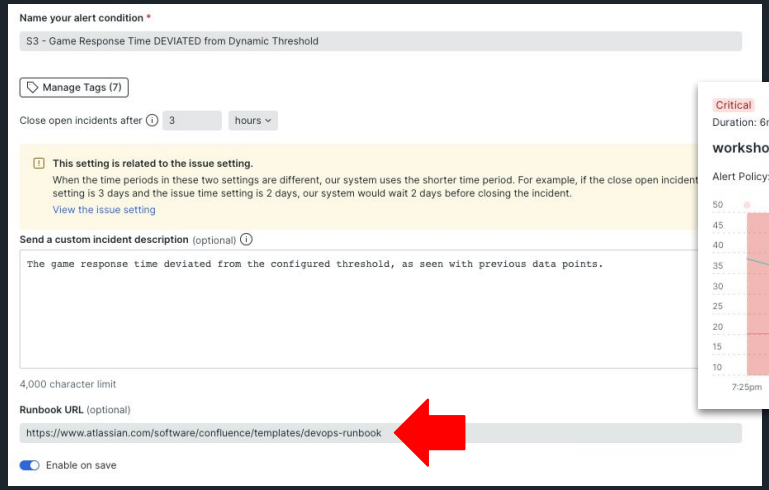
- The runbook must describe remediation steps to follow and who to involve or escalate to.
- Includes identifying the individuals or teams responsible for implementing each step and the channels that should be used to escalate the issue if necessary.
- Outline any preventive measures that can be taken to avoid similar incidents in the future.

## Answer

**https://www.atlassian.com/software/confluence/templates/devops-runbook**

# What is the current runbook URL?

## Challenge SEVEN





## Answer

**https://www.atlassian.com/software/confluence/templates/devops-runbook**

Improve Alert Response

new relic

# NRQL Condition tuning

**Many more features for refining how signal data is processed and evaluated.**

## Fine tuning

Configure aggregation windows, delay and streaming methods to reliably process your signal

## Gap filling

Reduce false alerts caused by sporadic data by specifying gap filling strategy

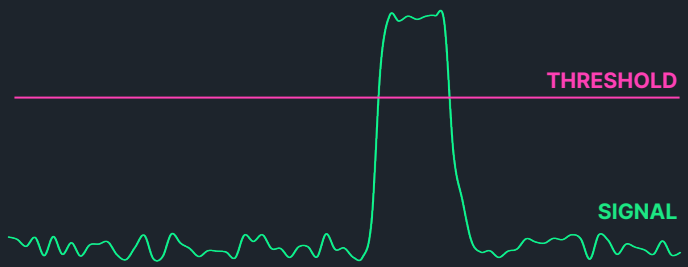
## Signal loss

What to do if a signal is no longer being received

**Improve Alert Response**

new relic

# Conditions & Threshold




**Some examples**

| Signal | Threshold |
|---|---|
| % Error rate | % error **> 5%** |
| Average basket value | median(value) **< $20** |
| Login attempts per user | uniqueCount(user) **> 10** |

## Signal Query

Defined by NRQL query including optional WHERE and FACET clauses.

Guided wizard available to help build.

**Additional non-NRQL conditions soon to be deprecated*

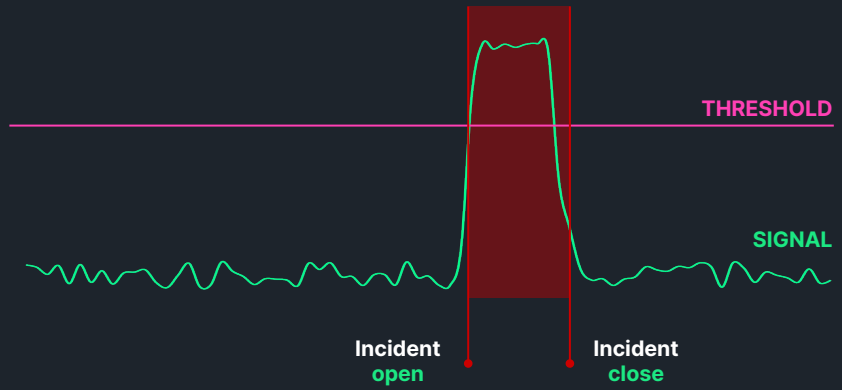
## Threshold

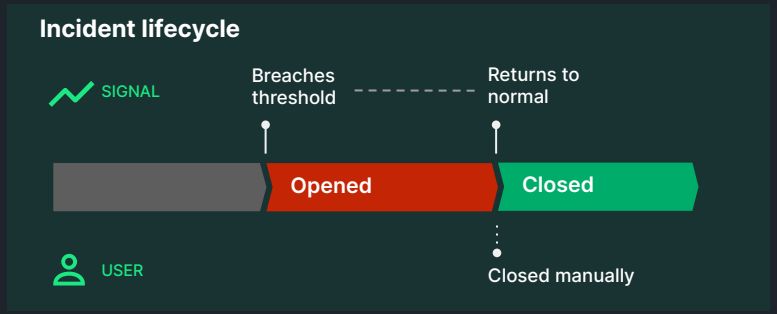
Value, which if breached for a defined time period, will trigger an incident to be opened.

Can be set for both **warning** and **critical**

**Improve Alert Response**

# Conditions & Threshold




- Incidents **open** when a **signal breaches the threshold** defined in a condition.
- Incidents **close automatically** when the signal is no longer in breach.
- There will be some **latency** between signal breaching and incident opening based on condition settings.
- Incident will open for **each signal facet**
- Incidents can be **manually closed** by user

**Improve Alert Response**

new relic

# 16 Challenges to Learn about AQM

**Alert Storm & Fatigue**

- M.E.L.T
- NRQL - FACET

**Bonus!**
**Post Mortem**

**Improve Alert Response**

- Error Inbox
- Naming
- Tags
- Anomaly Detection
- Runbook URL

**Improve Alert Quality**

- Issue Preference
- Notification Method
- Enrichment
- Notification Integration
- Muting Alerts

**Improve Alert Maintenance**

- Alerts Overview UI
- AQM Dashboard
- Mean Time to Close
- O11y as Code

**Improve Alert Quality**

new relic

# What are the issue preference options?

## Challenge EIGHT

- Creating a policy for each separate destination or audience needs to receive a notification.
- Consider grouping by entity, service, or technology to match the focus of your teams.
- Issues determine when you're notified about incidents disrupting your business. These incidents occur when they meet your alert conditions.

# What are the issue preference options?

## Challenge EIGHT

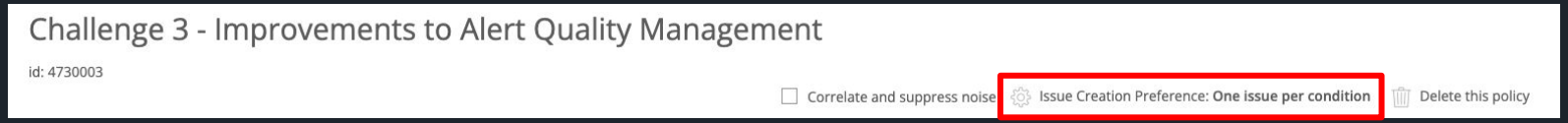
- Creating a policy for each separate destination or audience needs to receive a notification.
- Consider grouping by entity, service, or technology to match the focus of your teams.
- Issues determine when you're notified about incidents disrupting your business. These incidents occur when they meet your alert conditions.

## Answer

**One issue per condition**

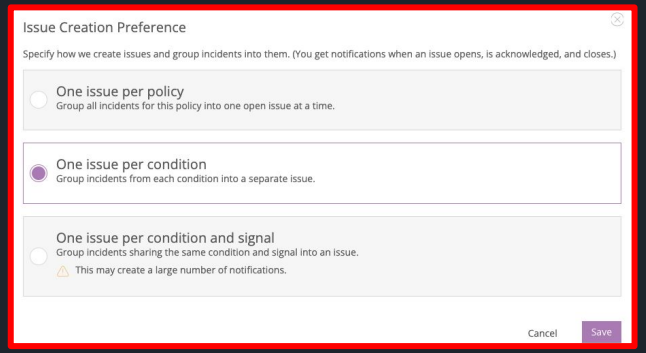
# What are the issue preference options?

## Challenge EIGHT




**Important!**
Ask your team 2 important questions when deciding your issue preferences:

Q1 - Do we want to be notified every time something goes wrong?
Q2- Do we want to group all similar notifications together and be notified once?




## Answer

**One issue per condition**

**Improve Alert Quality**

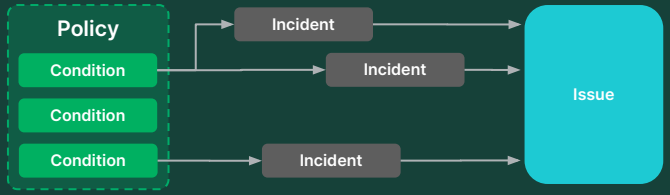
new relic

# Incident Groupings

FEWER ISSUES

MORE ISSUES

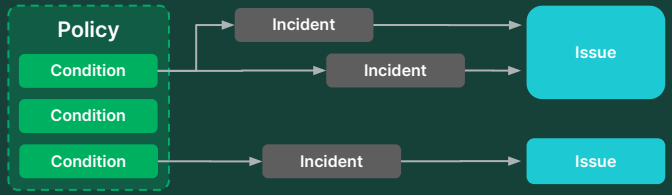
## 1 Per Policy




Only one issue will be open at a time for the **entire policy**.

- Requires immediate action and closing the issues to be effective

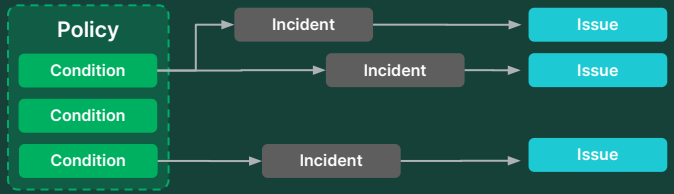
## 2 Per Condition




One issue will be open at a time for **each condition** in your policy.

- Useful for policies containing conditions that focus on entities that perform the same job

## 3 Per Incident



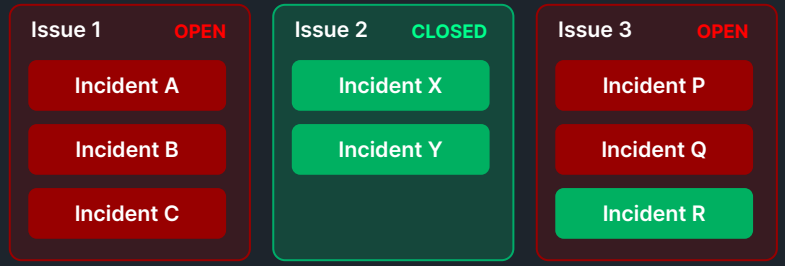

An issue will be created for **every incident** of **each condition** in your policy.

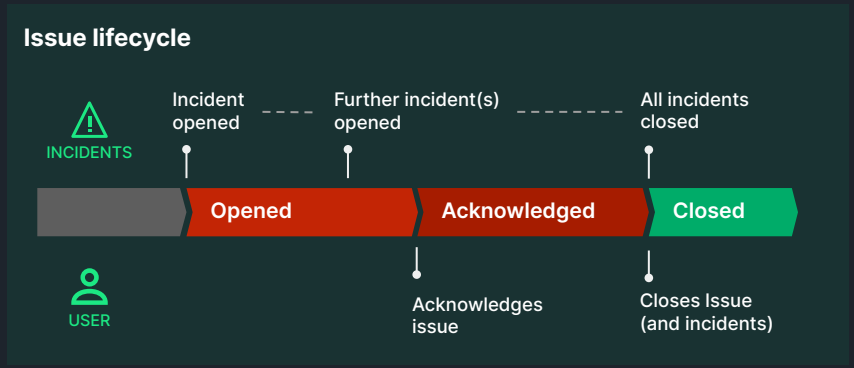
- Useful if you need to be notified of every violation or if you have an external system where you want to send alert notifications

**Improve Alert Quality**

new relic

# Incident Groupings




**Improve Alert Quality**

*"Incidents are the symptoms of a larger problem (the issue)"*

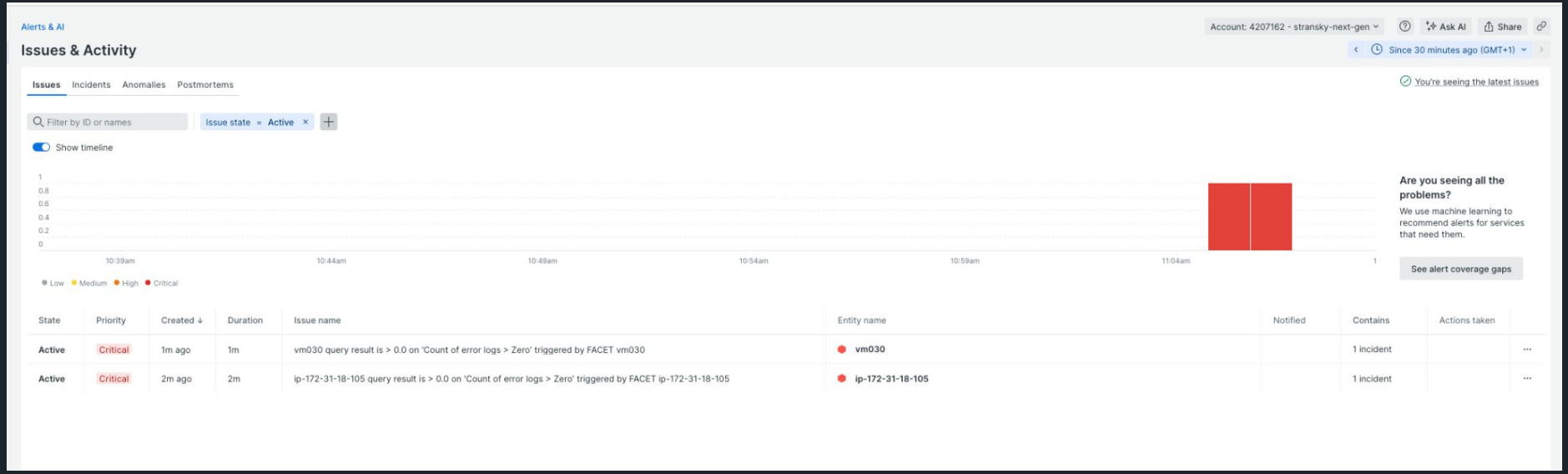
Issues **group incidents together**, reducing noise and driving notification workflows.

- Issues are **opened when incidents open**
- Issues can **contain multiple incidents**
  *based on policy preference or correlation decisions*
- Issues **close automatically** when all contained incidents have closed or if inactive
- Issues can be **manually closed**, which close all contained incidents.

new relic

# Quiz:  Alert on log messages with “error=”

# 2 Hosts received such messages




# Why are there 2 Issues ?



new relic

# 2 Hosts received log messages with “error=”

# Why are there 2 Issues ?

## NRQL Alert Condition with FACET:

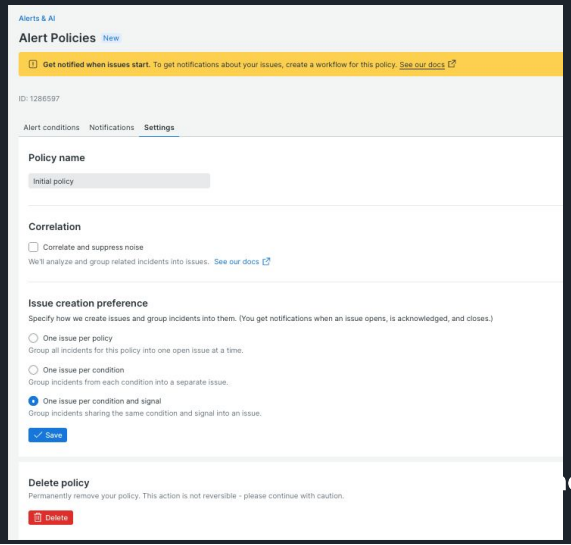
From Log Select count(*)

where message like ‘%error=%’

FACET hostname

## Alert Policy Issue Preference:

## One Issue per Condition and Signal






new relic

# 2 Hosts received log messages with “error=”


A)

Condition NRQL: From Log Select count(*)  where message like ‘%error=%’

Policy Preference: One Issue per Condition and Signal

How many incidents ?
How many issues ?

B)

Condition NRQL: From Log Select count(*)  where message like ‘%error=%’ FACET hostname

Policy Preference: One Issue per Condition

How many incidents ?
How many issues ?



new relic

# 2 Hosts received log messages with “error=”


**A)**

**Condition NRQL: From Log Select count(*) where message like ‘%error=%’**

**Policy Preference: One Issue per Condition and Signal**

**How many incidents ? ⇒ 1**
**How many issues ? ⇒ 1**

**B)**

**Condition NRQL: From Log Select count(*) where message like ‘%error=%’ FACET hostname**

**Policy Preference: One Issue per Condition**

**How many incidents ? ⇒ 2**
**How many issues ? ⇒ 1**



new relic

# The alerts send to, using which method?

## Challenge NINE

- Define alert categories, expectations for handling their notifications, and a unique destination across your organization
- For example, proactive to Slack to notify before the incident occurs; reactive to PagerDuty to detect and notify of an ongoing incident; or informative to Jira.

# The alerts send to, using which method?

## Challenge NINE

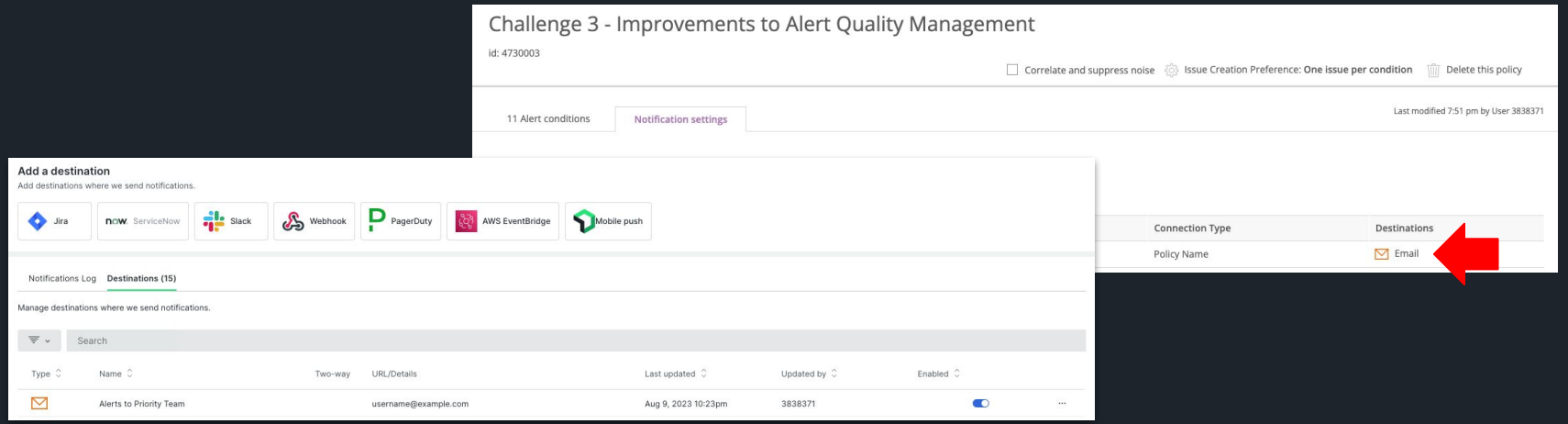
- Define alert categories, expectations for handling their notifications, and a unique destination across your organization
- For example, proactive to Slack to notify before the incident occurs; reactive to PagerDuty to detect and notify of an ongoing incident; or informative to Jira.

## Answer

**Email**

# The alerts send to, using which method?

## Challenge NINE




## Answer

### Email

**Improve Alert Quality**

new relic

# How do you enrich alerts in New Relic?

## Challenge TEN

- Enrichments can give additional context on alert notifications by adding NRQL query results to them. The workflows enrich tool is similar to the query builder.
- Triage your alert notifications faster by providing additional metrics specific to the issue.

# How do you enrich alerts in New Relic?

## Challenge TEN

- Enrichments can give additional context on alert notifications by adding NRQL query results to them. The workflows enrich tool is similar to the query builder.
- Triage your alert notifications faster by providing additional metrics specific to the issue.

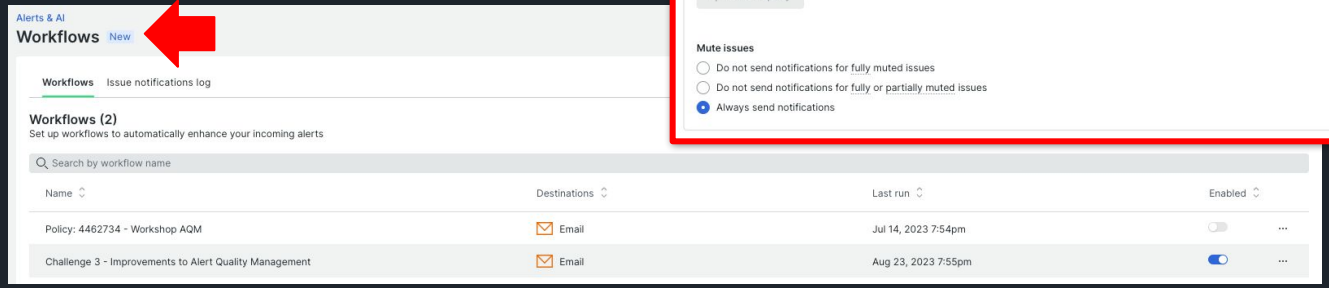
## Answer

### Alert Workflow Enrichment

**Improve Alert Quality**

# How do you enrich alerts in New Relic?

## Challenge TEN


## Answer

### Alert Workflow Enrichment

**Improve Alert Quality**

new relic

# Where can I find notification integrations?

## Challenge ELEVEN

- Add a responsible team as this team will be in charge of handling the first notification.
- Destinations are where we send notifications about your New Relic data. A destination is a unique identifier for a third-party system that you use.
- Destination settings contain the connection details to integrate with third-party systems and can be used across a variety of tools in New Relic.

# Where can I find notification integrations?

## Challenge ELEVEN

- Add a responsible team as this team will be in charge of handling the first notification.
- Destinations are where we send notifications about your New Relic data. A destination is a unique identifier for a third-party system that you use.
- Destination settings contain the connection details to integrate with third-party systems and can be used across a variety of tools in New Relic.

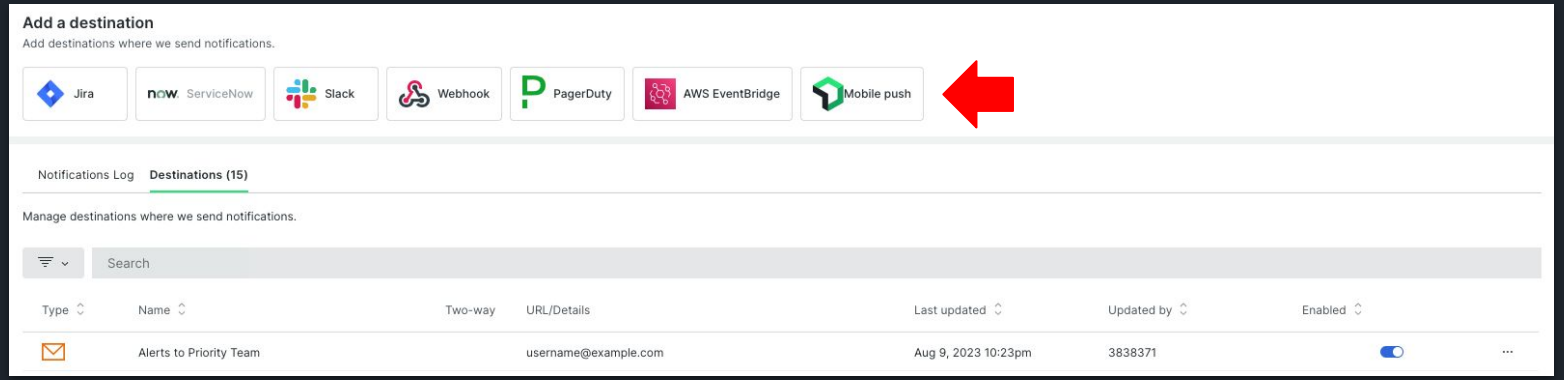
## Answer

**Alert Destination**

**Improve Alert Quality**

# Where can I find notification integrations?

## Challenge ELEVEN




## Answer

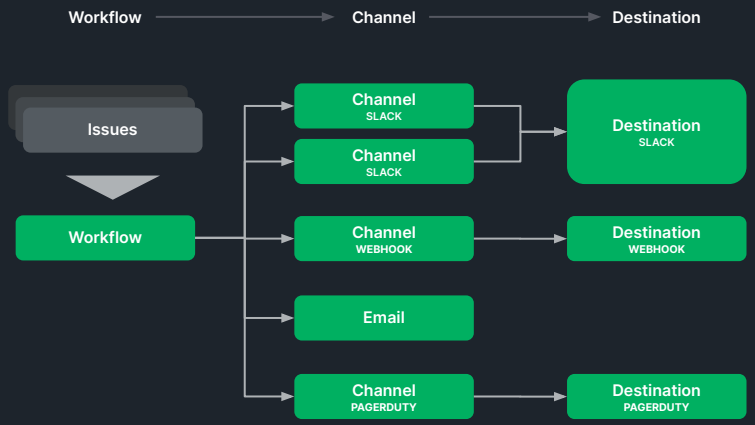
### Alert Destination

**Improve Alert Quality**

new relic

# Notification Triage and Routing

- Control **when** you want to receive notifications about issues to the right channel
- **Notify correct teams** based on issue context
- Channels offer comprehensive **payload templating** options
- **Enrich notifications** with additional context from other data points




Channels allow **multiple different message payloads** to be sent to **multiple destinations**

**Improve Alert Quality**

new relic

# How do I mute alerts?

## Challenge TWELVE

- Alerts sends out timely notifications when your system is having problems. You can use muting rules to stop being bombarded by messages that don't need your attention.
- Muted alerts will still gathers data on those incidents. Muting rules don't interfere with the alerts process and are applied at the point right before a notification is sent.

# How do I mute alerts?

## Challenge TWELVE

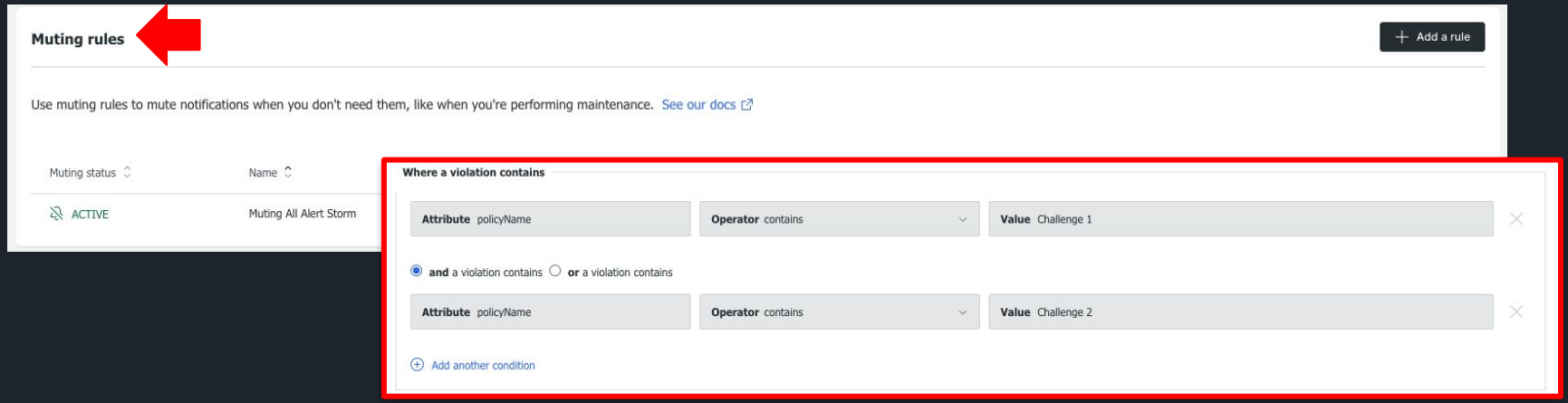
- Alerts sends out timely notifications when your system is having problems. You can use muting rules to stop being bombarded by messages that don't need your attention.
- Muted alerts will still gathers data on those incidents. Muting rules don't interfere with the alerts process and are applied at the point right before a notification is sent.

## Answer

**Challenge 1 & Challenge 2**

# How do I mute alerts?

## Challenge TWELVE




## Answer

### Challenge 1 & Challenge 2

**Improve Alert Quality**

new relic

# 16 Challenges to Learn about AQM

**Alert Storm & Fatigue**

- M.E.L.T
- NRQL - FACET

**Bonus!**
**Post Mortem**

**Improve Alert Response**

- Error Inbox
- Naming
- Tags
- Anomaly Detection
- Runbook URL

**Improve Alert Quality**

- Issue Preference
- Notification Method
- Enrichment
- Notification Integration
- Muting Alerts

**Improve Alert Maintenance**

- Alerts Overview UI
- AQM Dashboard
- Mean Time to Close
- O11y as Code

**Improve Alert Maintenance**

new relic

# Policy with the highest amount of alerts?

## Challenge THIRTEEN

- Schedule a periodical review of alert conditions - use the Alerts overview page to check the incidents created and decide the action to do.
- We recommend you to tag the condition with the last review date, which will allow you to identify obsolete alerts.

# Policy with the highest amount of alerts?

## Challenge THIRTEEN

- Schedule a periodical review of alert conditions - use the Alerts overview page to check the incidents created and decide the action to do.
- We recommend you to tag the condition with the last review date, which will allow you to identify obsolete alerts.

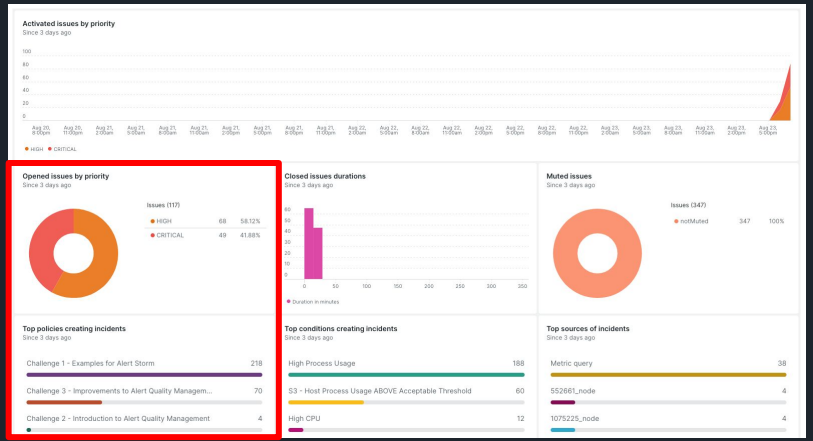
## Answer

**Top policies creating incidents**

**Improve Alert Maintenance**

# Policy with the highest amount of alerts?

## Challenge THIRTEEN




## Answer

### Top policies creating incidents

**Improve Alert Maintenance**

**Important!**
On the Alerts & AI Overview page, you'll find a consolidated view of your current alerts incidents. The Issues & activity page has views of your active issues, recent incidents, and anomalies.

**Top policies creating incidents**
A chart of the top five policies generating the most incidents. Click a policy name to filter the whole page by that policy's incidents.

**Top conditions creating incidents**
A chart of the top five conditions generating the most incidents. Click a policy name to filter the whole page by that condition's incidents.

**Top sources of incidents**
A chart of the entities generating the most incidents. If available, it uses the entity name. Otherwise, it will use the event type. Click an entity to filter the whole page by that entity's incidents.

new relic

# How do I install the AQM dashboard?

## Challenge FOURTEEN

- Review the alert quality dashboard - focus on reducing the number of nuisance incidents so that you focus only on alerts with true business impact.
- This reduces alert fatigue and ensures that you and your team focus your attention on the right places at the right times.

# How do I install the AQM dashboard?

## Challenge FOURTEEN

- Review the alert quality dashboard - focus on reducing the number of nuisance incidents so that you focus only on alerts with true business impact.
- This reduces alert fatigue and ensures that you and your team focus your attention on the right places at the right times.

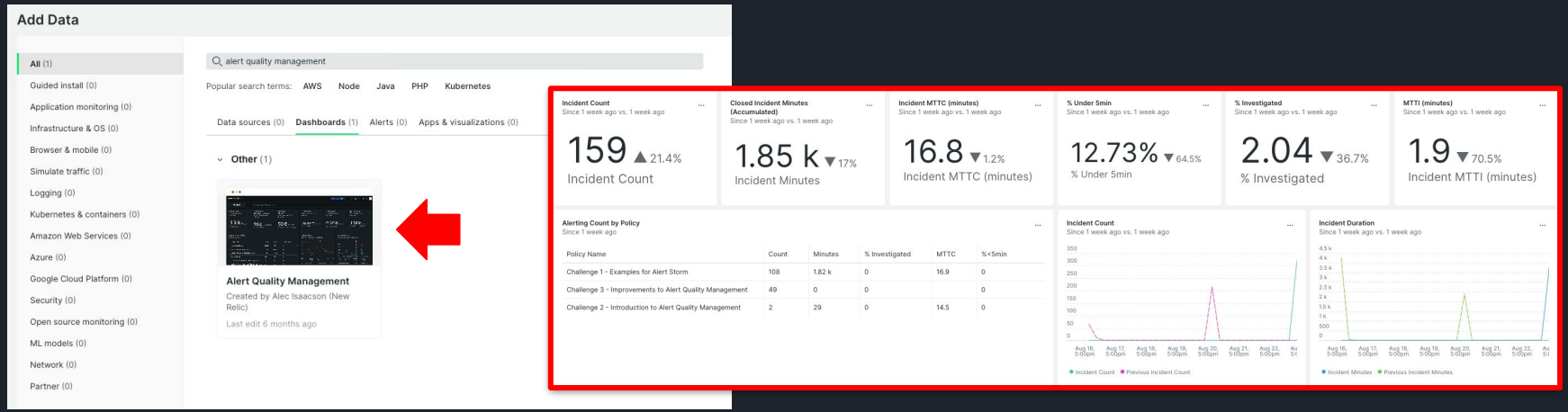
## Answer

**Add Data and search for Alerts Quality Management**

**Improve Alert Maintenance**

# How do I install the AQM dashboard?

## Challenge FOURTEEN




## Answer

### Add Data and search for Alerts Quality Management

**Improve Alert Maintenance**

new relic

# Exact NRQL needed to find out MTTC?

## Challenge FIFTEEN

- Mean Time to Close (MTTC) - This is the average duration of incidents within the period of time measured. You want this number to be as low as possible.

# Exact NRQL needed to find out MTTC?

## Challenge FIFTEEN

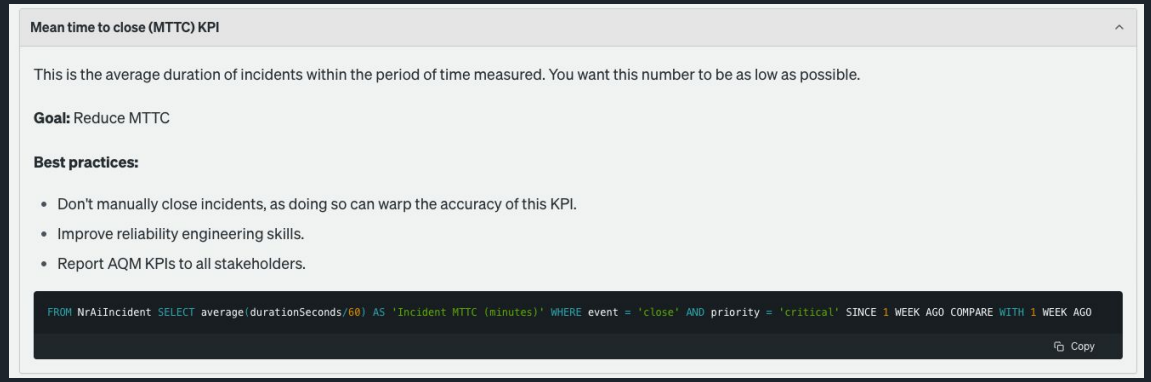
- Mean Time to Close (MTTC) - This is the average duration of incidents within the period of time measured. You want this number to be as low as possible.

## Answer

**FROM NrAiIncident SELECT average(durationSeconds/60) AS 'Incident MTTC (minutes)' WHERE event = 'close' AND priority = 'critical' SINCE 1 WEEK AGO COMPARE WITH 1 WEEK AGO**

# Exact NRQL needed to find out MTTC?

## Challenge FIFTEEN




https://docs.newrelic.com/docs/tutorial-create-alerts/manage-alert-quality/#volume

## Answer

**FROM NrAiIncident SELECT average(durationSeconds/60) AS 'Incident MTTC (minutes)'**
**WHERE event = 'close' AND priority = 'critical' SINCE 1 WEEK AGO COMPARE WITH 1 WEEK AGO**

# How do you use Terraform?

## Challenge SIXTEEN

- Terraform is a popular infrastructure-as-code tool built by HashiCorp to provision all kinds of infrastructure and services, including New Relic dashboards and alerts.
- Automate your alert creation using Terraform where you can prevent undocumented changes and clear traceability by installing Terraform, read the getting started guide and visit the New Relic official guide.

**Improve Alert Maintenance**

# How do you use Terraform?

## Challenge SIXTEEN

- Terraform is a popular infrastructure-as-code tool built by HashiCorp to provision all kinds of infrastructure and services, including New Relic dashboards and alerts.
- Automate your alert creation using Terraform where you can prevent undocumented changes and clear traceability by installing Terraform, read the getting started guide and visit the New Relic official guide.

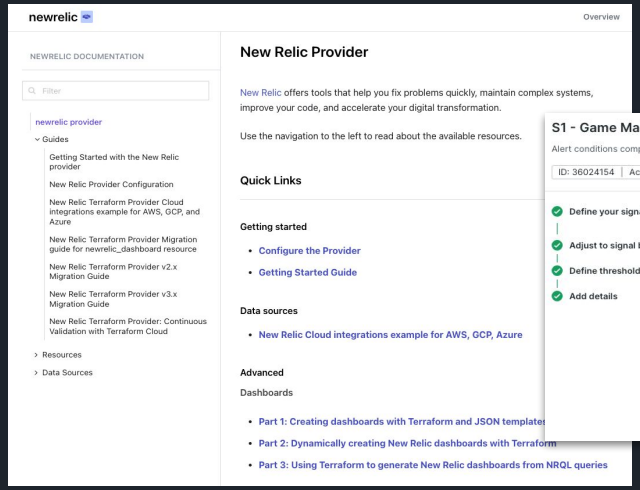
## Answer

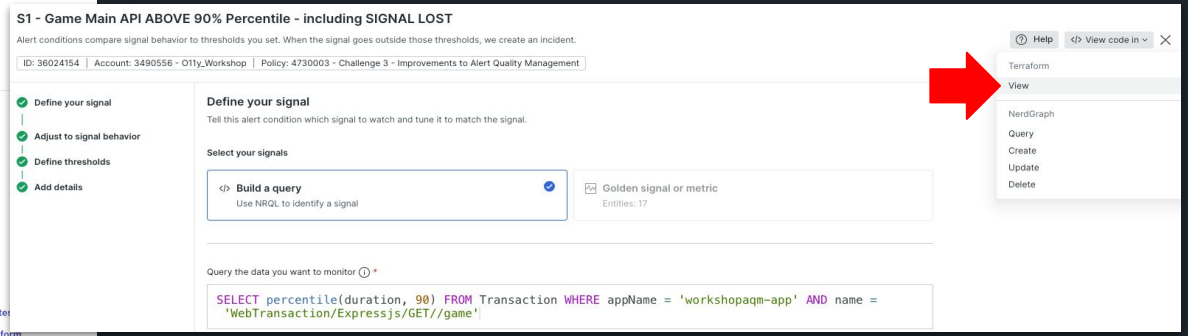
**Terraform cmd examples as seen in the workshop**

**Improve Alert Maintenance**

# How do you use Terraform?

## Challenge SIXTEEN




You can also view the TF code via the UI

## Answer

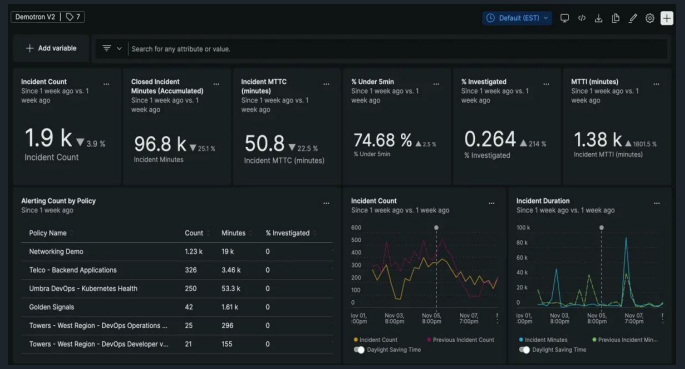
### Terraform cmd examples as seen in the workshop

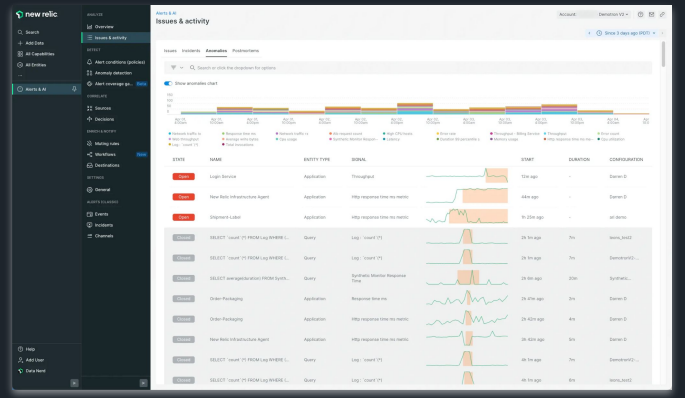
**Improve Alert Maintenance**

new relic

# Alert Quality Management

- Strategies to focus on **reducing the number of nuisance incidents** so that you focus only on alerts with **true business impact**
- Ensure that **fewer, more valuable**, incidents are created for the right purpose
- Includes **strategies, best practice** and **tooling** for measuring and analysing **alert quality**



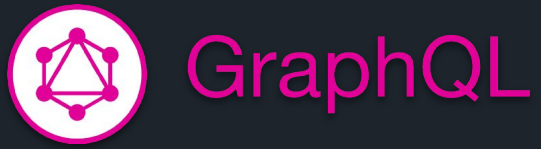

**Improve Alert Maintenance**

new relic

# Observability as Code

- Use graphQL (or Terraform) to **manage alerts programmatically**
- Allows for better **auditing** and **change control**
- Build workflows to **generate alerts automatically** from configuration

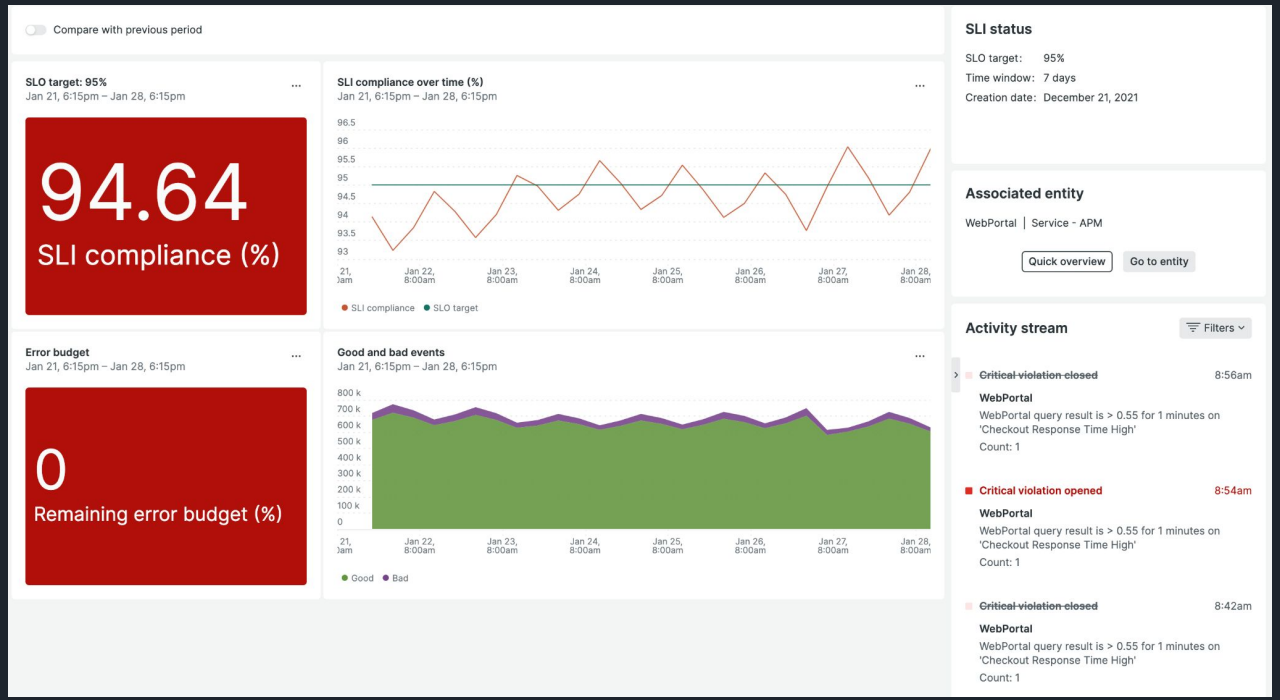

- Scale
- Stability
- Reusability
- Automation
- Compliance
- Security
- Innovation



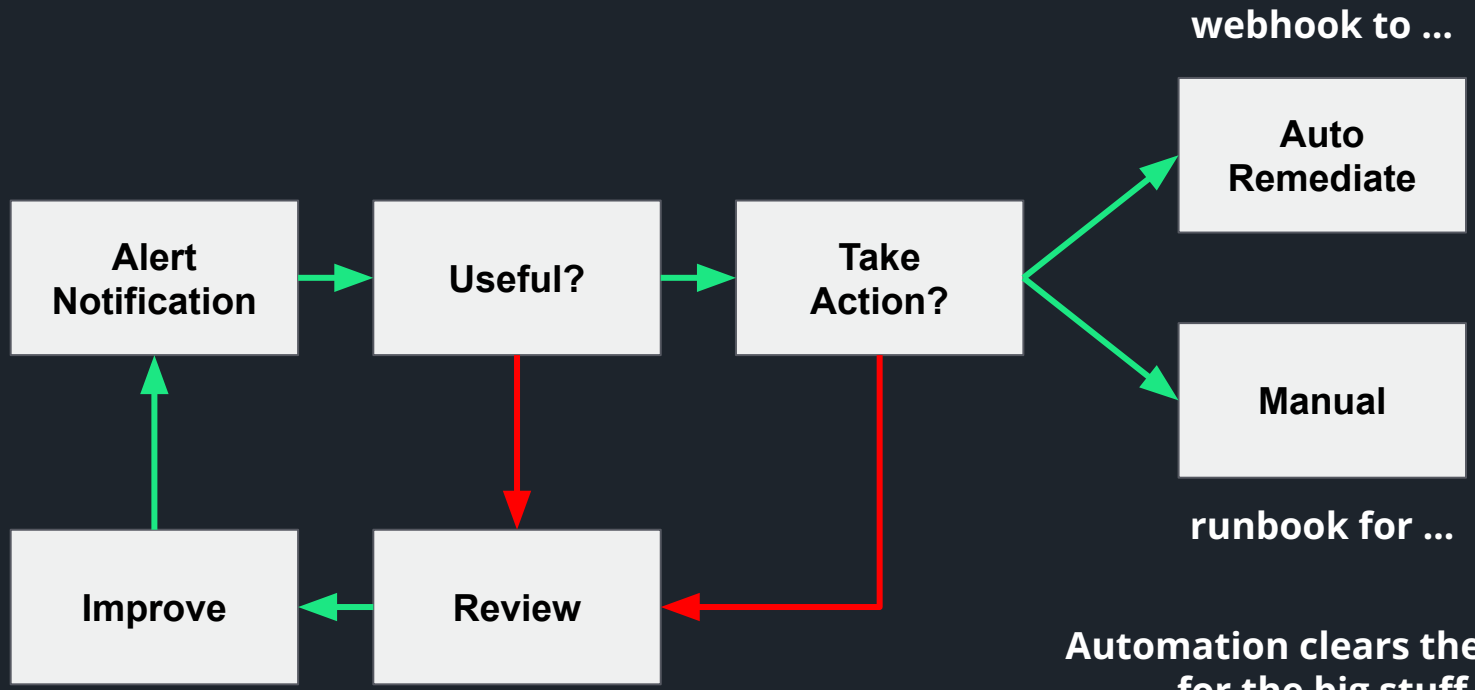

**Improve Alert Maintenance**

# Going Next Level

## SRE based Alerts

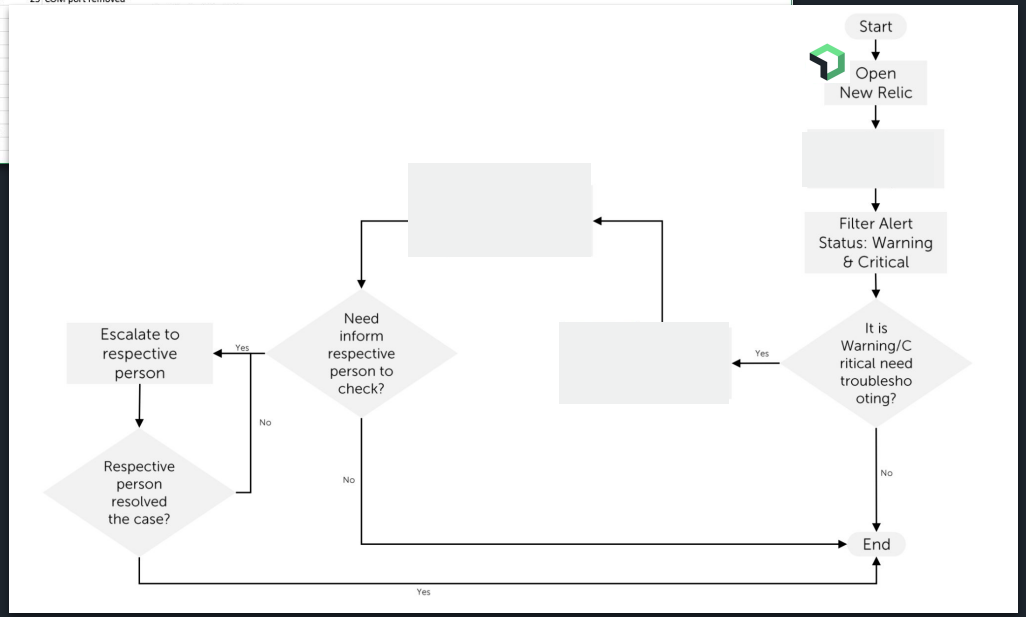


# End Goal - Alert Workflow



# Insights - Standardization - Automation

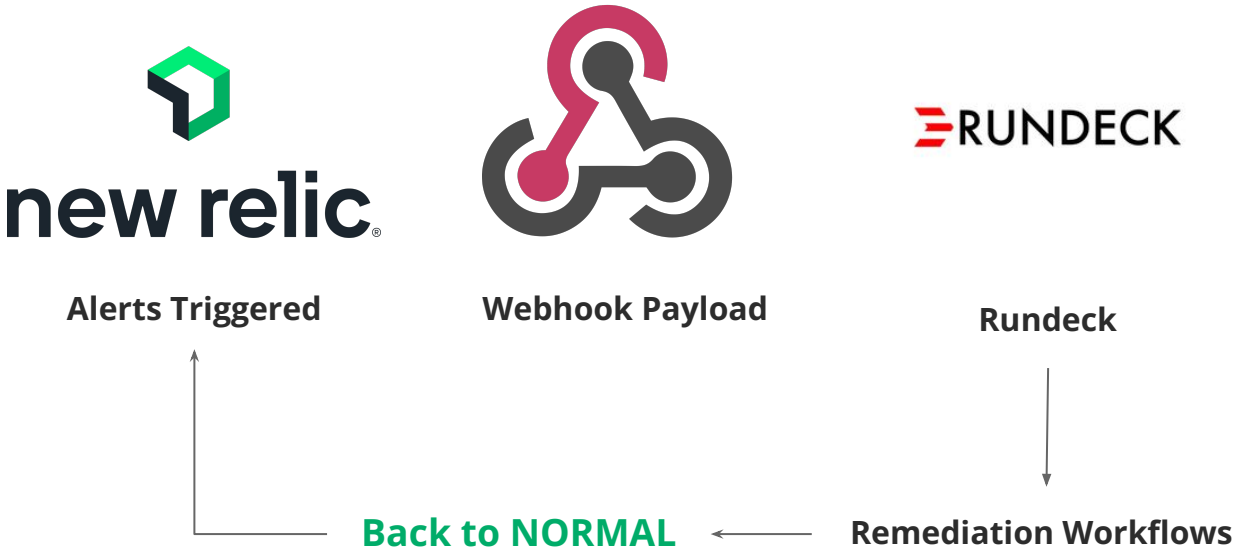
| Message | Logs | |
|---|---|---|
| Secondary 1 error: serial comm error; (16) The directory cannot be removed. | 1954 | COM port removed |
| Printer warning: printer not enabled | 722 | Printer is not enabled (expected) |
| Reactor warning: resetting event loop (thread id=4294967295)... | 538 | Reactor watchdog timeout, many possible reasons. |
| NozzleInEvent error: hose 12.1 not valid | 178 | Invalid nozzle ID event. May be caused by misconfiguration |
| Application error: failed to get hose 0 | 127 | Couldn't locate hose 0 object |
| Pump 1: state change from PriceChanging to SystemError: (7 -> 1) | 59 | Price change timed out |
| Pump 5: Gilbarco Protocol: Data error received when sending preset limit | 56 | Dispenser did not accept the last multibyte command. If this is happening often it may indicate a dispenser communication issue and should be investigated further. |
| Pump 5: state change from SystemError to NotResponding: (1 -> 5) | 52 | Dispenser is not responding |
| Hose 7.0 warning: invalid state 3 for FuelNotFlowingEvent event | 45 | Requires additional context to identify the cause. |
| Pump 2: error: failed to set current delivery | 41 | Requires additional context to identify the cause. |
| NozzleOutEvent error: hose 13.1 not valid | 37 | Couldn't locate hose 0 object |
| NozzleInEvent error: hose 13.1 not valid | 37 | Couldn't locate hose 0 object |
| Secondary 8 error: serial comm error; (1460) This operation returned because the timeout period ex | 25 | COM port removed |
| Timer -1, CPump warning: StartTimer failed to acquire mutex | | |
| ObjectBackup warning: failed to backup object 102, CDeliveryStack (retries: 9) | | |
| Deserialize error: bad buffer alignment | | |
| Pump 5: delivery 11 error: delivery totals lost - clear required | | |
| Timer 19, CDelivery, warning: 300000 ms delay failed to signal; signalling now... | | |
| Pump 7: delivery 78 error: Preauth overrun, limit 80.00, total amount 80.23 | | |
| Timer 15, CPump, warning: 20000 ms delay timed out prematurely; rescheduling for 3601937 ms | | |
| Pump 5: error: unexpected PumpLocked event; state Authorizing aborted | | |
| ObjectStore error: invalid set at location 3088; deleted... | | |
| ObjectStore error: failed to move file pointer, offset 1296, origin 0: file ../journal/ObjBackup.obs not | | |
| An error occurred processing <CommandReq> element | | |
| Pump 15: delivery 36 error: Preauth overrun, limit 80.00, total amount 85.43 | | |



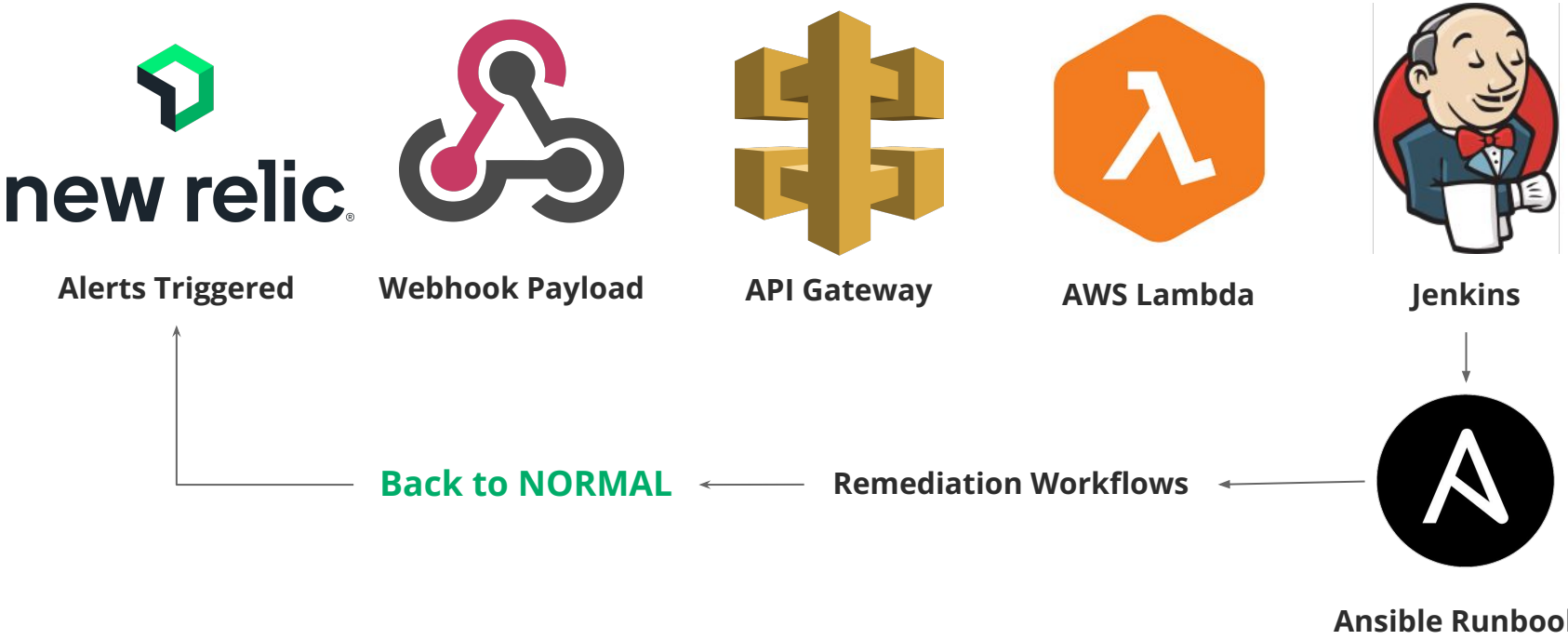

Match **PROBLEMS**,
with the right **WORKFLOW**

Once the right workflow is in place, optimization is **EASY**

# Examples - Self Remediation



# Examples - Self Remediation



# Alert Quality Management

## Resources to help you get better

- Alerts Best Practices
- Manage Your Alert Quality
- Alerting Concepts & Terms
- Issue Preference Options
- Anomaly Detection for Alerts
- Workflow Enrichment for Alerts
- Muting Rules
- Introduction to Terraform with New Relic
- New Relic Terraform Provider
- Hands On Lab with AQM


You can fix Alert Storm & Fatigue!



new relic