Proactive outage avoidance

300-level live demo script

Introduction

In this demo, I'll show you how Cloud Pak for Watson AIOps helps SREs and IT Ops teams proactively identify, diagnose, and resolve incidents across mission-critical workloads.

You'll see how Watson AIOps:

- Intelligently correlates multiple disparate sources of information, such as logs, metrics, events, tickets, and topology
- Condenses and presents all of this information in actionable alerts instead of large quantities of unrelated alerts
- Resolves problems within seconds to minutes of being notified using Watson AIOps' automation capabilities

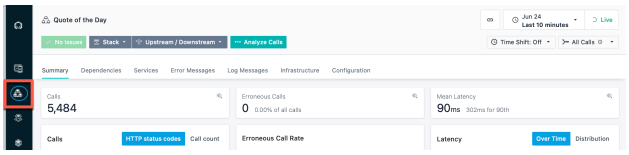
We will be using an application called Quote of the Day, which is a content delivery app that serves up random quotations. This will serve as a proxy for any type of application. The application is built on a microservices architecture, and the services are running on Kubernetes.

1 - Simulating a failure

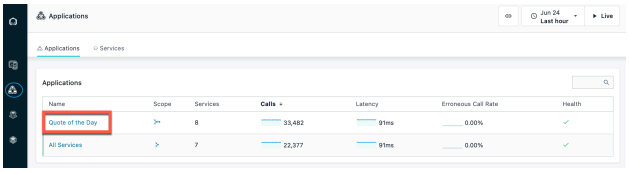
1.1 - Review the golden signals of the Quote of the Day (QotD) application

Narration
Let's examine the current health of the Quote of the Day application.

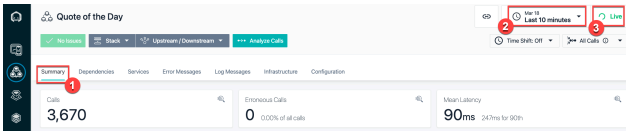
Action 1.1.1
- Navigate to **Instana**, and click the **Applications** icon.



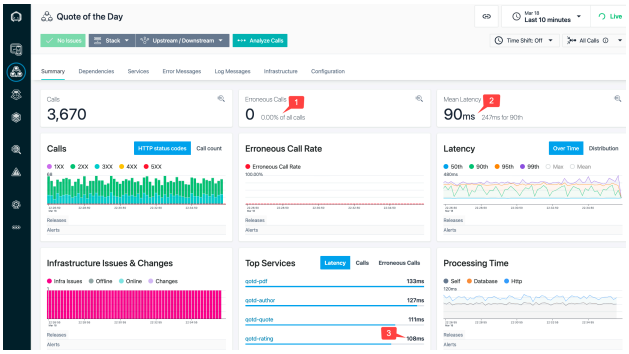
Action 1.1.2
- Click the **Quote of the Day** application.



Action 1.1.3
- Click the **Summary** tab (1). Set the time period to **Last 10 minutes** (2), and click **Live** (3).

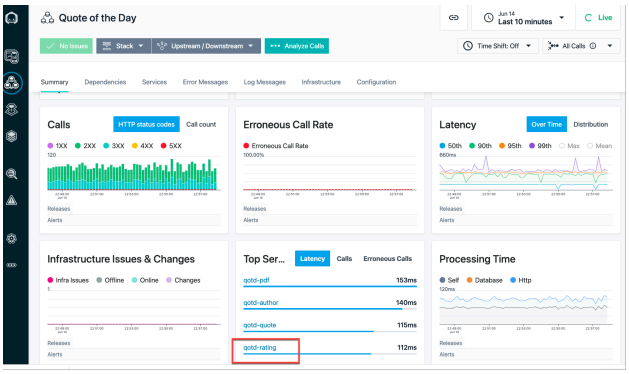


Action 1.1.4
- Show the application is healthy by pointing out the erroneous call rate is 0 (1), the mean service latency is 90 ms (2), and the average latency of the Rating (**qotd-rating**) service is 108 ms (3).
- **Note**: The numbers on your screen may vary slightly.

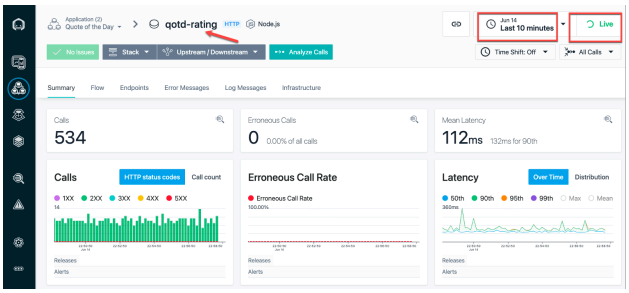


Narration
Let’s take a closer look at the qotd-rating service.

Action 1.1.5
- Click the **qotd-rating** service.



Action 1.1.6
- Examine the golden signals of the **qotd-rating** service. Ensure that the time window is set to **Last 10 mins** and that the refresh is set to **Live**.



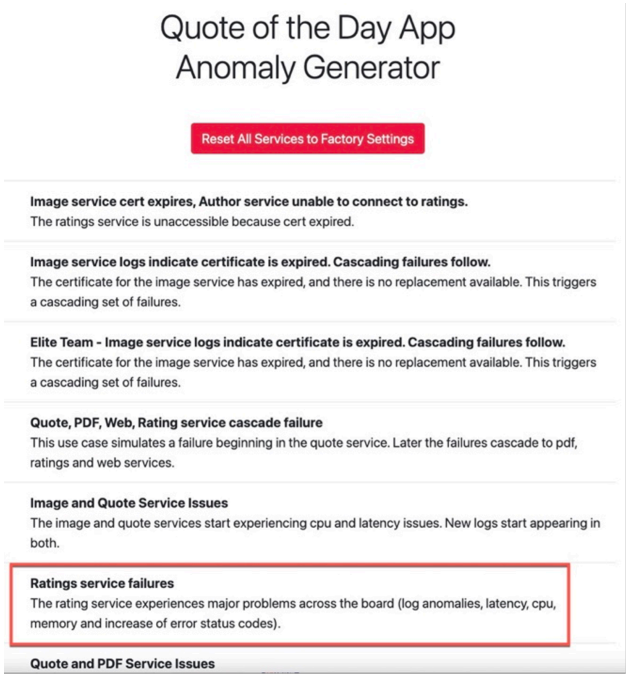
Narration
The qotd-rating service is healthy. The erroneous call rate is 0, and the mean service latency is sub-second.

We are now ready to inject a few failures and observe how AIOps detects and helps quickly resolve incidents.

1.2 - Navigate to the anomaly generator and inject failures

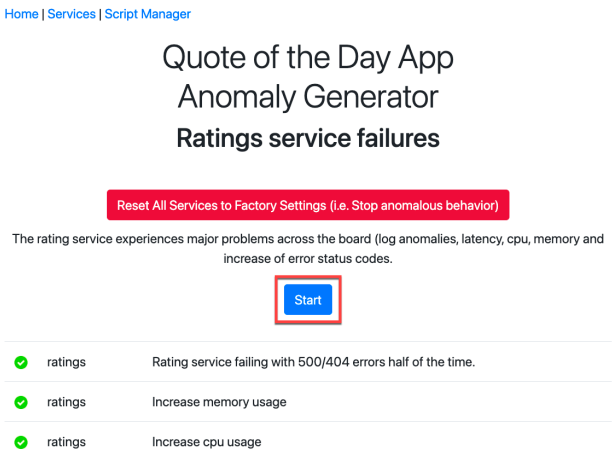
Narration
To see how this all works, I'm going to generate a few real failures in our application.

Action 1.2.1
- Select the **Quote of the Day Anomaly Generator** tab. Click **Ratings service failures**.



Narration
The anomaly generator web application is designed to simulate a variety of failures.

Action 1.2.2
- Click **Start**.

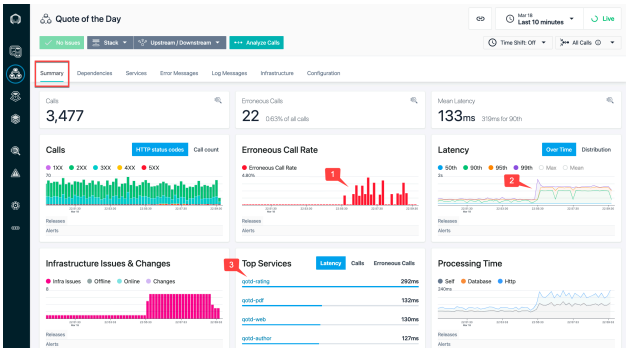


Narration
The green check marks indicate that the various failures are now underway. The various anomalies represent real-world performance issues that applications typically face, such as high CPU usage, memory leaks, or network congestion.

Now, let's examine the current state of the Quote of the Day application and understand the impact of the failures. Observe the increase in the number of calls, the error rate, latency, and CPU processing time.

Action 1.2.3

- Navigate back to the **Instana** tab. Click the **Summary** tab and point out the trends in the primary golden signals.
- Point out the increases in both the erroneous call rate (1) and the mean service latency (2). Also, notice that in the Top Services chart, the **qotd-rating** service is now at the top of the list (3).



Narration
Without Watson AIOps, we would get all sorts of alerts and notifications from multiple sources when a problem occurs.

As we'll see in a moment, with Watson AIOps, all of this information gets correlated and presented in one place. Watson AIOps also includes recommendations for how to resolve the incident. We can take action directly from the notification and resolve the incident quickly.

2 - Getting notified of an emerging problem

2.1 - AIOps packages the notification along with relevant details as a story

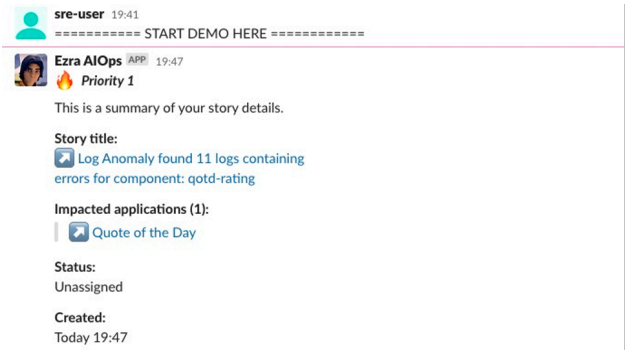
Narration
Notifications are now appearing in Slack. We're using Slack in this demo, but Watson AIOps also integrates with Microsoft Teams.

Watson AIOps formats the notifications into a "story" using AI to correlate events, metrics, alerts, and logs. Each story brings together the various notifications for all the affected services by the same underlying issue. Imagine if each piece of data presented in the story was a separate notification – we'd quickly be inundated with alerts.

The story is like a home base for action when a problem arises. Instead of manually correlating things across multiple different tools, it's all right here immediately when the notification is received.

In addition to providing a highly contextualized view of the incident, it enables us to jump to other tools to explore further details. This eliminates tool silos and helps us restore service faster.

Action 2.1.1
- Navigate to the tab running **Slack** and examine the incoming story.



Narration
This story is telling us there's an emerging problem with the Rating service, which is one of the microservices in our Quote of the Day application.

In the background, the AI and ML algorithms of Watson AIOps have done the work for us. It shows which services are affected and presents us with a curated view of relevant information: the events and alerts that are indicative of the symptoms of this problem, anomalies that Watson AIOps has found in the log files, and similar incidents that have occurred in the past. This way, we can see how they were successfully resolved. We'll explore each of these components in more detail.

3 - Determining which services caused the problem

3.1 - Analyze the story

Narration
Based on its intelligent correlation of logs, metrics, and traces, Watson AIOps derives probable causes of the failure.

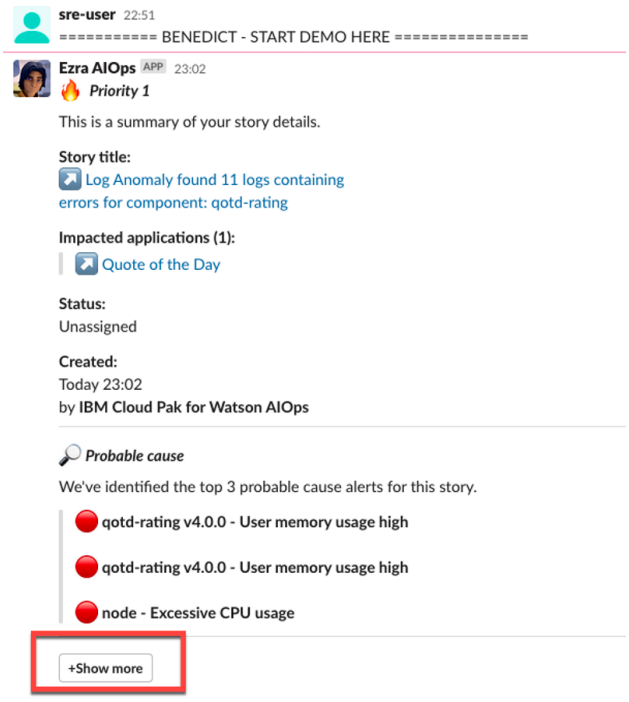
As we can see in the ChatOps notification here, there is a high-priority issue emerging in the QotD application.

Watson AIOps presents a list of top three probable causes of this incident: a log anomaly in the qotd-rating service, an excessive CPU usage condition, and increasing service call response times.

As more information comes in, Watson AIOps correlates related events and updates the existing story in real time.

Let's dive deeper into the story to examine the supporting details.

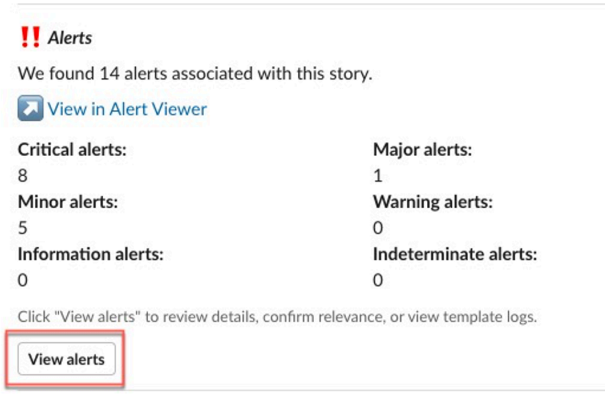
Action 3.1.1
- Click **Show more**.



Narration
Let's unpack the details behind the developing story. We need to find out where the issue began, so we can prevent it from causing cascading failures across the components of the application.

Instead of having to hop between tools to look for alerts, we can see them right here from the notification. Watson AIOps has determined that these events are related, and it provides an explanation for how it determined the relationships. We can see that there are two groups of events based on related resources and the timing of the events.

Action 3.1.2
- Click **View alerts** to examine the details behind these alerts.



Narration
We can inspect the grouped events right here without searching for them in another tool.

It looks like the memory and CPU on the Rating service increased significantly. This is causing a significant slowdown in the response times on both the Rating and Web services.

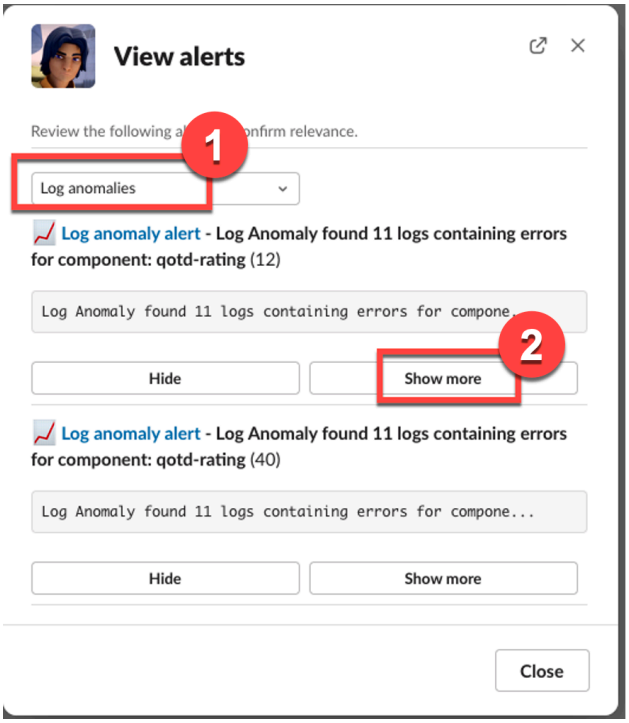
Based on this information, it seems that the Rating service is the source of the issue. Let's get a bit more detail – this time from the log files.

Instead of needing to go to Kibana and manually sort through the hundreds or thousands of log entries that come in every minute, Watson AIOps has found several anomalies in the log files and presented them here.

During normal operations, Watson AIOps continually trains on the log files of the application and monitors for deviations from that baseline. We can see that the anomalies are occurring on the Rating service, which fits with what we saw in the alerts.

Action 3.1.3

- In the **View alerts** pop-up window, select **Log anomalies** (1) and then **Show more** (2).



Narration

Watson AIOps gives us additional context on the anomaly. In this case, the 'error_log' anomaly is telling us that Watson AIOps has never seen this type of log entry before (hence the 'unknown') and that the log message indicates there is some type of error. Watson AIOps is not only looking at the statistical frequency of the type of log, but it is also using Natural Language Processing (NLP) to analyze the content of the log message to give additional context (in this case that there's likely an error).

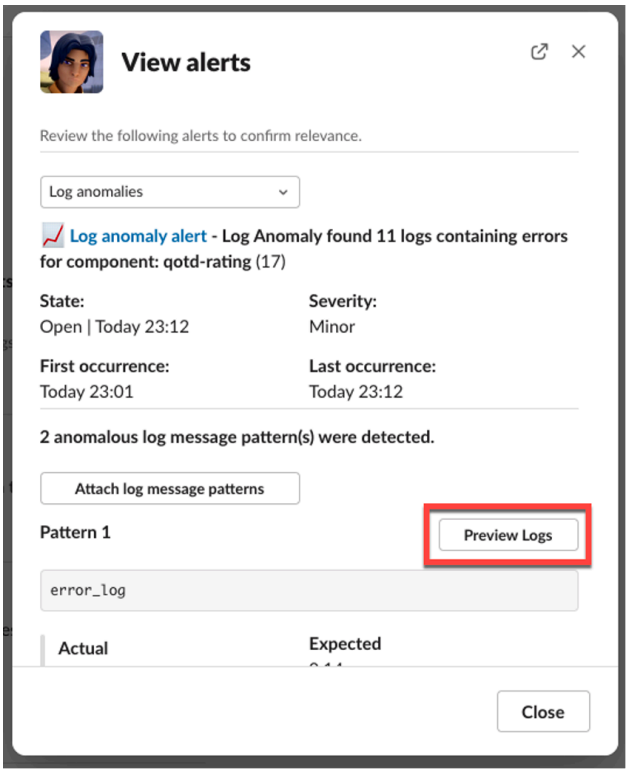
Watson AIOps also explains why the anomaly was flagged. It expected to see zero of this type of log, but it actually saw four.

Now we know that there is an unfamiliar log coming from the Rating service, and it's indicating an error.

This further reinforces what we saw with the alerts - it looks like the Rating service is likely the root cause of this problem.

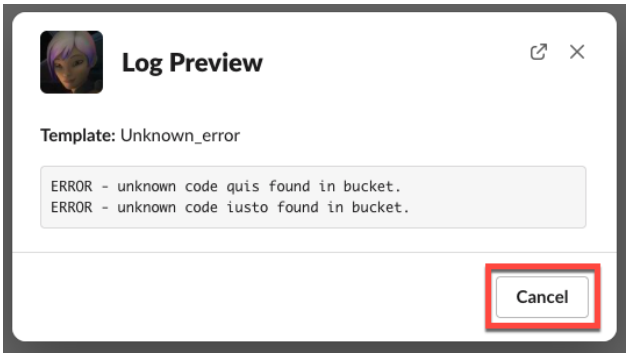
Action 3.1.4

- Click **Preview Logs** of **Pattern 1**.
- **Note**: Pattern 1 displays '**error_log**.'



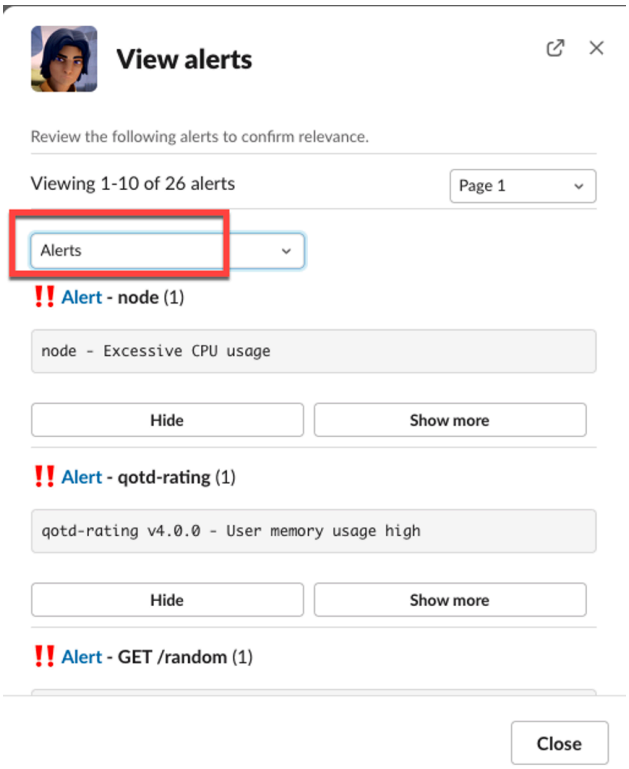
Action 3.1.5

- Click **Back** in the **Log Preview** window.



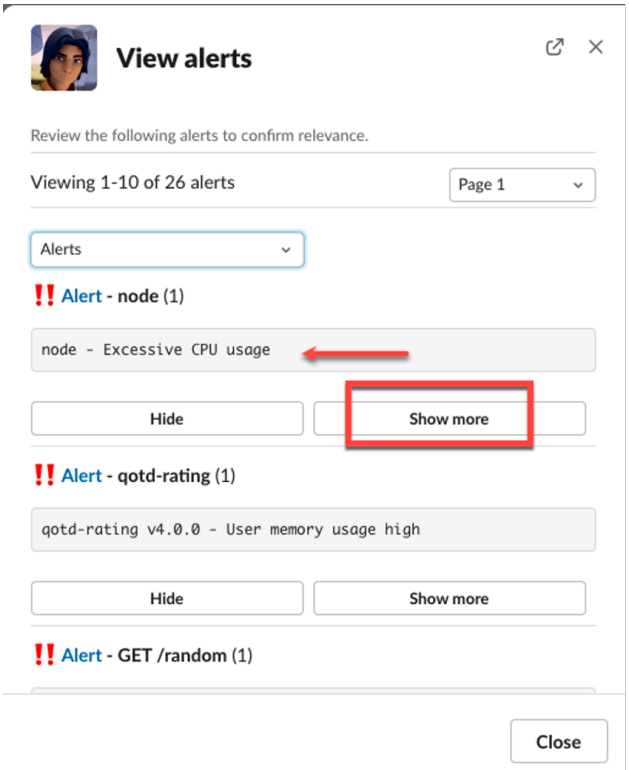
Action 3.1.6

- In the **View alerts** pop-up window, select **Alerts** from the drop-down menu options.



Action 3.1.7

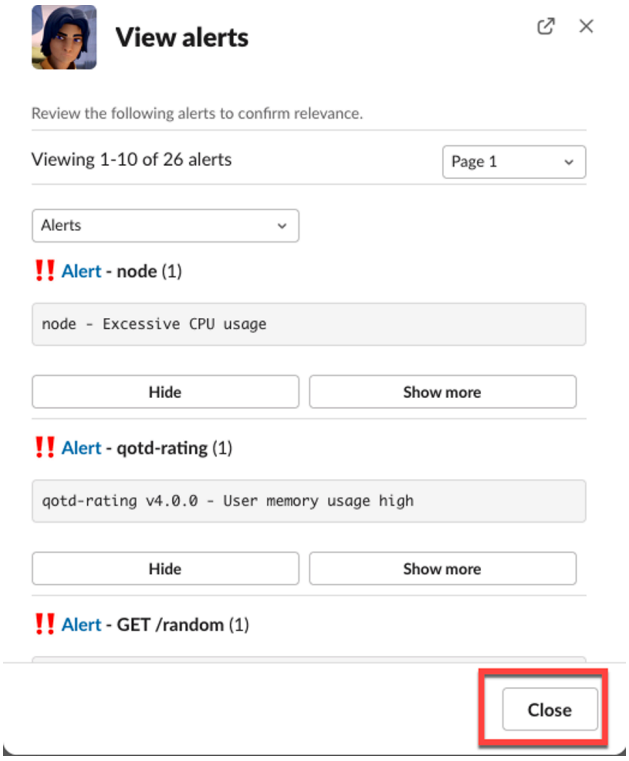
- Click **Show more**.



Narration
Watson AIOps has detected that the qotd-rating service is consuming an excessive amount of CPU and memory resources.

Under the covers, Watson AIOps is comparing the CPU and memory utilization to the known and learned behaviors of the application. Since it is seemingly straying from the expected behavior, Watson AIOps is flagging the anomaly and correlating it to the current story.

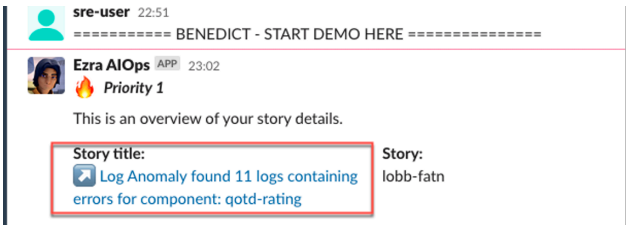
Action 3.1.8
- Click **Close** to exit the **View alerts** pop-up window.



Narration
We have reviewed the log and metric anomalies that have been correlated as part of this story by Watson AIOps.

Now that we have a high-level understanding of the emerging incident and the probable causes, we need to focus on resolving the issue as quickly as possible.

Action 3.1.9
- Click the **Log Anomaly found 11 logs containing errors for component: qotd-rating** story title to navigate to the Watson AIOps console.



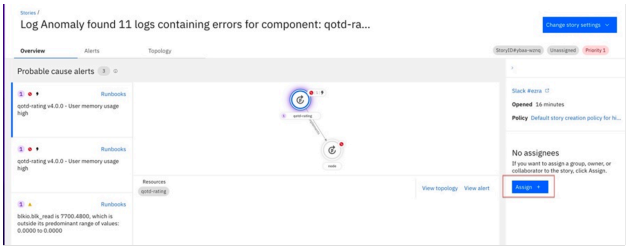
4 - Fixing the problem and restoring the service

4.1 - Take ownership of the incident

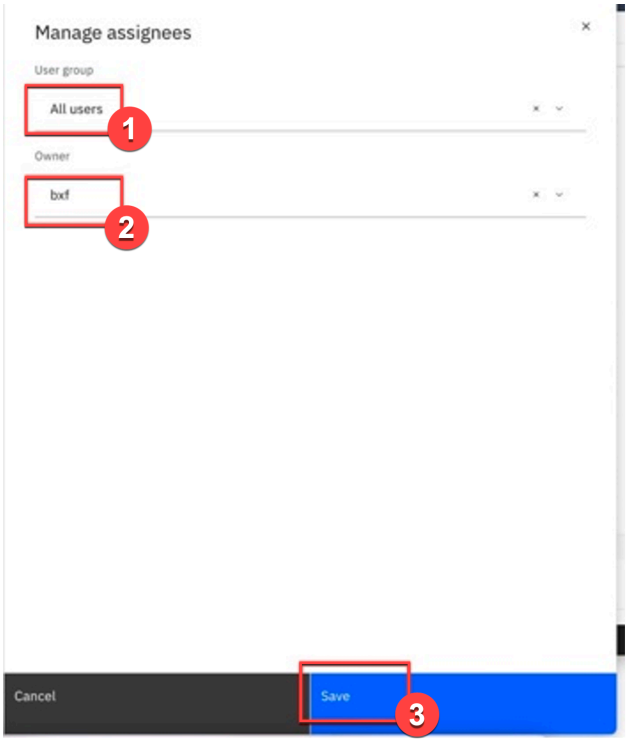
Narration
The Watson AIOps console enables us to get to the details underlying the story. It allows us to examine the specific alerts and understand the topological relationships between the impacted application components. It also helps accelerate incident resolution by recommending specific runbooks that could potentially resolve the current incident.

Before we progress with the incident resolution process, it is important to take ownership of the story. This ensures accountability and enables the collaboration and communication necessary to accelerate the resolution of the incident.

Action 4.1.1
- To take ownership of the incident, click **Assign**.



Action 4.1.2
- Set the **User group** to **All users** (1), and set the **Owner** to your CoC ID (2). Click **Save** (3).



Action 4.1.3

- Validate that you are now the owner of the incident.

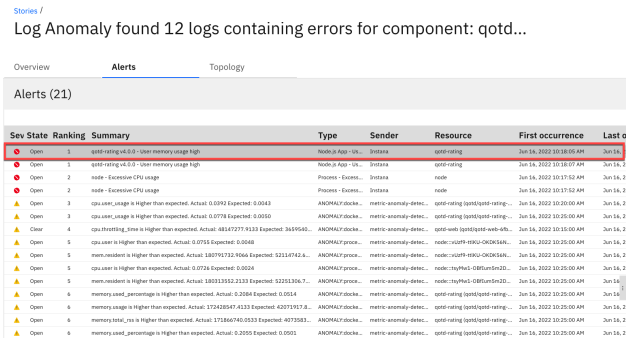
Narration

Let's proceed to fix the problem and restore the service.

Now that we have a high-level understanding of the incident, let's take a quick look at the topology of the application to understand which components are impacted and their relationships with other services in the application. The component dependencies will help us better understand and assess the potential impact of cascading failures.

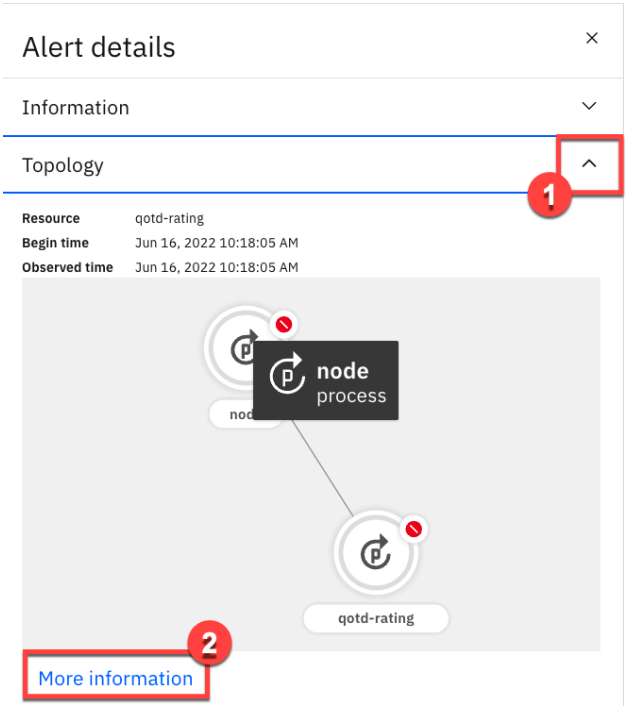
Action 4.1.4

- In the **Alerts** tab, select the first alert in the list that confirms a critical alert in the **qotd-rating** service.



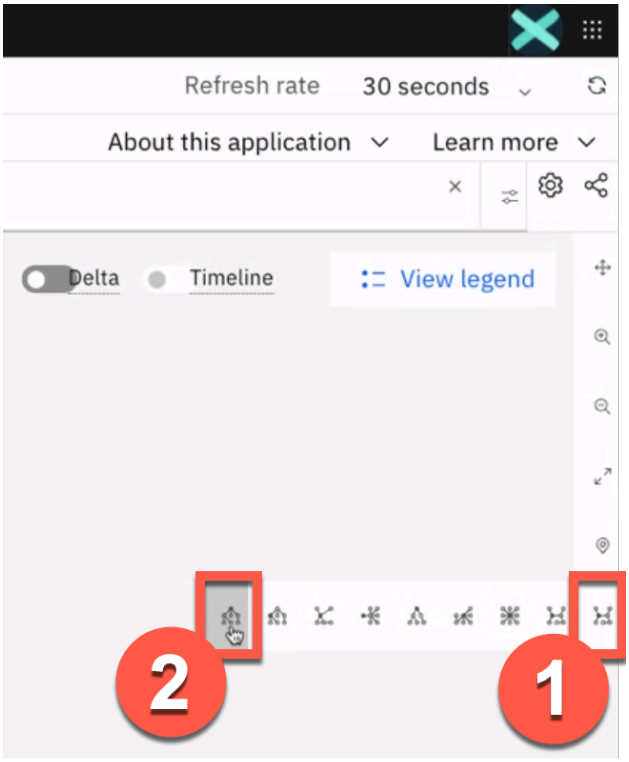
Action 4.1.5

- Expand **Topology** by clicking the **down** arrow (1). Click **More information** to get an expanded view of the topological relationships and dependencies of the qotd-rating service.



Narration
Resources are related to each other, and these relationships are represented in the graph as a typed arc.

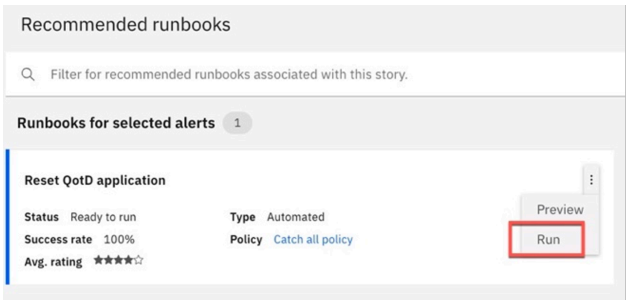
Action 4.1.6
- Click the **Hierarchy** icon (1), located on the right side, to change the topology orientation to an **Inverted tree** (2).



Narration
These discovered topological relationships are used when grouping alerts. If two resources are directly or closely related and they are exhibiting an alert, they get grouped together in the same story.

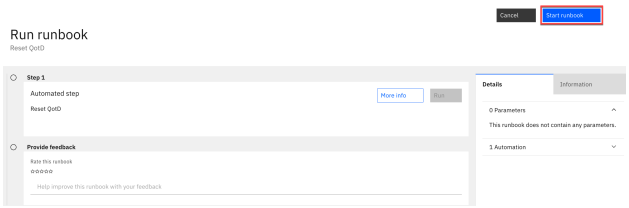
The qotd-rating service is implemented with two pods. Therefore, there are two separate resources for pods, containers, processes and Node.js runtimes.

Action 4.1.7
- Navigate back to the **Overview** tab to execute the recommended runbook. In the **Recommended runbooks** section, click **Run** from the pop-up menu.



Action 4.1.8

- Click **Start runbook**.



Action 4.1.9

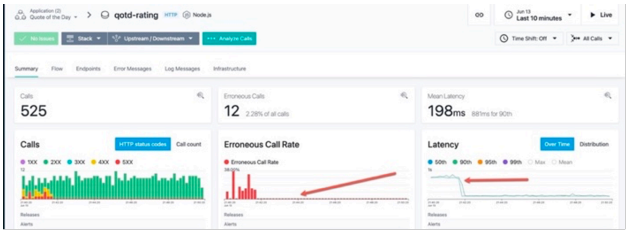
- Click **Run** to start the execution of the runbook.
- **Note**: Do **NOT** click **Complete** yet. That will take place later.

Narration

Before we mark the runbook as complete, let's confirm that the erroneous call rate is decreasing and that the service latency is improving.

Action 4.1.10

- Navigate to the **Instana** tab. Validate that there are improvements in the erroneous call rate and latency.

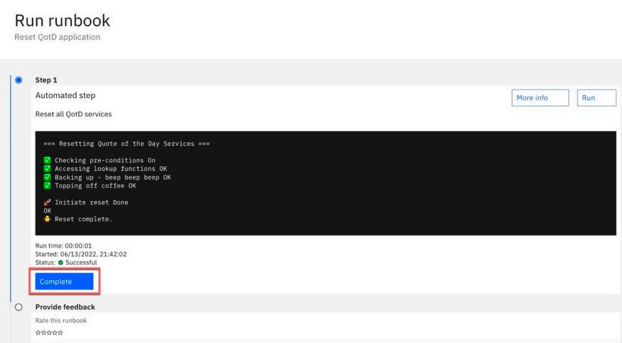


Narration

Over time, the golden signals of the qotd-rating service will indicate a 100% success rate and a sub-second service latency.

Action 4.1.11

- Click **Complete** to mark the successful execution of the runbook.



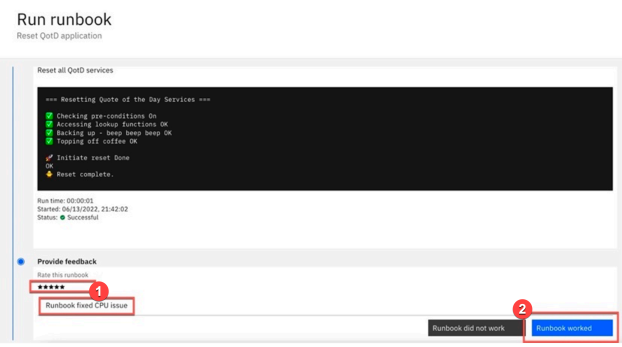
Narration

We can rate the runbook to document the fact that it was successful. This will be helpful for the next person who uses the runbook, as well as for providing feedback to the author of the runbook.

This enables a collaborative approach to organically improve incident resolution over time.

Action 4.1.12

- Provide feedback and rate the runbook (1). Then, click **Runbook worked** (2).

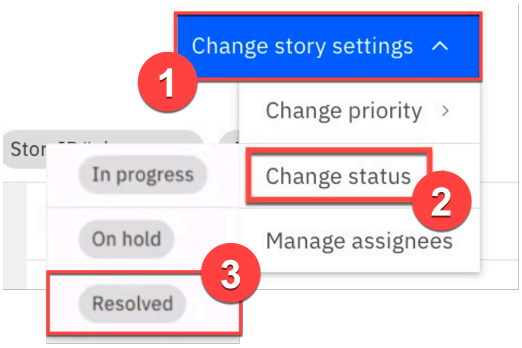


5 - Closing the incident

5.1 - Mark the story as resolved

Narration
Now that the incident is resolved and the service is restored, let's update the status of the incident.

Action 5.1.1
- Expand **Change story settings** (1) and click **Change status** (2). Mark the status of the incident as **Resolved** (3).



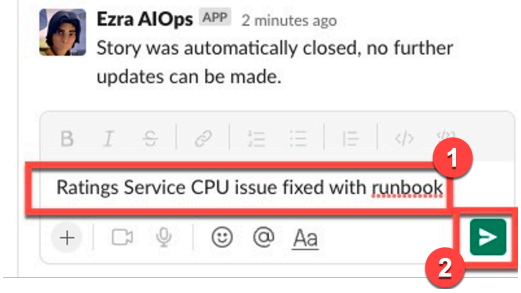
Narration
Watson AIOps will mark the incident as resolved and then change the status to 'Closed' automatically.

After Watson AIOps has marked the incident as 'Closed,' let's navigate to Slack to communicate the successful resolution of this incident to the broader team.

Action 5.1.2
- Navigate to **Slack**.

Narration
Watson AIOps adds a reply to the relevant Slack thread, indicating the formal closure of the story.

Action 5.1.3
- Optionally add any relevant comments to communicate with the broader team regarding this incident (1). Then, click the green **Send** icon.



Summary

In this demo, we have demonstrated how Watson AIOps enables you to avoid business-impacting outages by applying AI to data gathered from your existing disparate tools. It helps you quickly determine the probable causes of a failure and proactively alleviate outages, therefore minimizing the business impact of these IT issues on revenue or client experiences.

The anomaly detection capabilities alert you early to potential issues, enabling the SREs to quickly take remedial actions. The intelligent event analytics examine logs, metrics, tickets, and topology and provide a useful correlation that would otherwise be very challenging. Lastly, the AI-driven remediation accelerates problem resolution and significantly reduces the mean time to repair.

Thank you for attending today’s presentation.