Troubleshooting customer complaints of checkout latency issues

You are a site reliability engineer working in a cloud-native environment that runs several microservices to support an online store. Your customer support team has received a number of complaints from customers who are experiencing checkout delays, so you need to troubleshoot and fix the issue.

Data required

Business service data

How to use Splunk software for this use case

1. Open Splunk Observability Cloud, navigate to the main menu, and click Splunk Real User Monitoring.

2. The Splunk Real User Monitoring home screen shows you data covering load times, activity, and front end errors. Scan through this information and look for problems. In this example, you can see that the /cart/checkout node latency is high. Click it to investigate further.

3. Looking at the /cart/checkout node, you can see there has been a latency jump. To look more closely, click SESSIONID.

4. From the trace view, you can identify requests with latency issues. In this instance, you can see a 1.73 second delay. To access more information in Splunk APM, click APM in the center row of this area.
5. Here, you can see the backend view of the trace. In this example you can see some errors in cartservice, checkoutservice, and paymentservice. At the bottom of this backend view, click the `frontend/cart/checkout` link under Workflow Name to access the service map.

6. Looking at the service map, check for long response times. In this example you can see latency issues running from the external client, through the frontend and checkoutservice, flowing through to the paymentservice. It's not immediately obvious that this problem is caused by errors stemming from the paymentservice, so you'll need to investigate further.
7. Click on paymentservice and click on one of the errors in the Request Rate area to get some example traces. Click the Trace ID for one of the traces to see more.

8. The waterfall view of this trace shows several errors. In this example it looks like checkoutservice is doing an exponential backoff, which results in the high latency bars you can see at the top of this area.

9. At this stage, you might want to double-check what environment, or environments, the errors are originating in. In the drop-down box under paymentservice, click Environment. This shows you the breakdown of errors in the environments you are running.
10. Next to the drop-down field you just changed to access Tag Spotlight, click **Spotlight**. Tag Spotlight shows you all the tags coming in for these spans so you can look for problems within these tags. In this example, you can see that version 350.10 is correlated with all the errors that are happening, indicating that this deployment might need to be rolled back.

11. Open Log Observer and click the + in the top toolbar. Click **Fields**, find **label_app** in this area, and click **paymentservice**. Then, filter for error logs in the bar chart. In this example you can see multiple serialization failures occurring.

12. To be completely sure that the new deployment is causing these errors, click **Visual Analysis** at the top of the screen and sort by version. Here you can see that version 350.10 is causing these serialization failures.
13. *(Optional)* Click one of the serialization failure events to copy the stacktrace and send it to your team so you can later merge a fix for the issue.

14. After you’ve rolled back the deployment, you can use Splunk Observability Cloud to double-check that the errors have stopped. Using the time picker at the top-left of the screen, click **Live Tail**. This mode allows you to see any errors that arise in real-time. As the deployment rolls out, you should see these errors stop, meaning that the issue is mitigated.

Next steps

Latency and errors in eCommerce environments can result in customers abandoning their planned purchases, with financial impact to your organization. Quickly identifying and fixing these issues results in fewer dollars lost and fewer frustrated customers.

The content in this guide comes from a .conf2020 talk, **Logging for Observability**, one of the thousands of Splunk resources available to help users succeed. In addition, these Splunk resources might help you understand and implement this use case:

- Use case: **Troubleshooting a high error rate alert on a checkout endpoint from APM**
• Use case: Assessing the financial impact of eCommerce checkout errors
• .Conf Talk: Splunking application performance: Traditional APM and beyond
• Docs: Analyze service performance with Tag Spotlight
• Tech Talk: Find root causes in minutes with the Splunk APM tag spotlight