Incident Sep. 27, 2022: Partial global sales outage due to network failure across multiple failure domains

Availability: **Approx. 50% of the global sales flow and 100% of the admin module were impacted.**

% of clients affected: **All clients were impacted.**

Duration of incident: **The global sales flow was impacted by 50% during 2h15min. The admin module was impacted by 100% during 3h48min**

Symptoms

Between 16h12 UTC and 19h56 UTC, our customers experienced high latency and timeout-associated errors in our administrative environment, aka VTEX Admin. Starting at 17h13 UTC, the same incident began to impact sales, affecting approx. 50% of orders per minute. We prioritized mitigating the incident on the sales flow and had our global orders return to normal levels at 19h28 UTC (2h 15min later).

Summary

We are constantly updating our systems to add new features for you and to prevent any security issues.

This Tuesday, September 27, 2022, latency for some of our systems increased rapidly. Our automated monitoring system alerted us something was not quite right. It is worth noting the systems in scope are spread across multiple clusters (i.e., different failure domains) that do not communicate with each other. They do not communicate with each other in order to prevent widespread outages.

Our incident response team quickly gathered to investigate what began as a VTEX admin-only outage. At first, we uncovered that we had not altered anything in the minutes and hours prior to the perceived impact. We then manually increased capacity at first, bypassing our autoscaling system. Retries were occurring, so we had an extra load on the platform at that point. This capacity increase process took some time and did not mitigate the problem.

Upon further investigation, we uncovered that part of our networking-related setup within those clusters was broken. Since it was not immediately clear what broke it (human operator error or automation), we attempted to reinstate the networking setup. It partially mitigated the incident. The networking setup recovery itself in Admin clusters but not in Store Clusters.

In parallel, our incident response team initiated a process to rollback this entire subset of our platform to a previously known healthy state — this involved much more than a software rollback. We performed an infrastructure rollback. This is what ultimately mitigated the issue after 2 hours and 15 minutes (17h13 UTC–19h28 UTC).

We have since then identified and fixed the automation issues that triggered the incident. In the next few days, we will perform a more detailed retrospective exercise within VTEX in order to strengthen our ability to prevent whole classes of incidents in the future, and once they occur, we will be better prepared to mitigate them faster and with reduced impact on you.

We are now entering a mode of much-reduced risk tolerance for upcoming customer campaigns and Black Friday week. We will continue to modernize our platform for you next.

We hope that the coordination described below gives our customers the peace of mind that we act during the incident with the rigor and engagement they deserve.

Timeline

[2022-09-27 16:29 UTC] We received an alert that latency was very high in one Admin cluster.

[2022-09-27 16:44 UTC] The incident was confirmed once our customer success (CX) team reported errors were in fact widespread on VTEX admin.

[2022-09-27 16:53 UTC] We recognized one of our multiple **admin clusters** had higher 5XX errors and high latency after an initial investigation. The other clusters were healthy at this point.

[2022-09-27 16:59 UTC] We identified the trigger; the cluster's network interface failed several nodes on one admin cluster.

[2022-09-27 17:09 UTC] We divided the team to work on different frontlines: (1) spinning up new clusters to move traffic from unhealthy ones; (2) investigating what happened with the network interface and trying to recover it; and (3) to find contributing factors, we began to investigate a possible IPs drain-out and if this could be causing the network interface failure.

[2022-09-27 17:13 UTC] We received the alert of degradation in the global order dashboard and identified the **first Store cluster** had its latency increased too. Given that, we changed the incident severity and started investigating the store cluster.

[2022-09-27 17:25 UTC] We found changes applied to the network configuration that inhibited the cluster's elasticity and noticed that all the clusters had the same issue. Given that, we performed a rollback to the previous configuration on all of them.

[2022-09-27 17:33 UTC] We discarded the possible IP drain-out.

[2022-09-27 17:38 UTC] We finished the spin-up process of the new cluster and ***prioritized*** sending Stores' traffic to it.

[2022-09-27 17:39 UTC] The **second Admin cluster's** latency increased, and we then had two unhealthy admin clusters.

[2022-09-27 17:47 UTC] We observed the first admin cluster started to recover after we fixed its network configuration.

[2022-09-27 17:53 UTC] The **second Store cluster's** latency increased, and we then had two unhealthy Store clusters.

[2022-09-27 18:00 UTC] We realized that our fix only recovered one cluster, the other three were still unhealthy. We started to investigate why. Meanwhile, we continued to move traffic

from one Store cluster to the new one and started the process of spinning up another cluster aiming to send all the Stores' traffic to healthy clusters.

[2022-09-27 18:38 UTC] The second cluster spinning-up process ended and we started to send Stores' traffic from the second unhealthy cluster to the new one. Since we still had an unhealthy admin cluster, we started a new cluster spinning-up process.

Between **[2022-09-27 18:38 UTC]** and **[2022-09-27 19:23 UTC]**, our focus was divided into: (1) carefully moving Stores' traffic for the two new clusters; (2) keep trying to recover unhealthy clusters after the network interface fix; and (3) find what triggered the issue in the network interface to avoid it from happening again.

[2022-09-27 19:23 UTC] We finished the process of moving traffic from the second unhealthy Store cluster to a healthy one. Immediately after this, we observed the global sales rapidly coming back to the normal level.

[2022-09-27 19:28 UTC] We observed that the global orders were back to the normal level.

[2022-09-27 19:29 UTC] We observed that the second Admin cluster started to recover after we fixed its network configuration.

[2022-09-27 19:32 UTC] We declared that our clusters were functional again and started the monitoring process. Meanwhile, we pinpointed the code automation that triggered the network interface failure and removed it from the change management workflow to avoid the issue from happening again.

[2022-09-27 19:56 UTC] We declared that Store clusters were fully operational and we started investigating which applications were increasing the latency of our Admin clusters.

[2022-09-27 20:34 UTC] We adjusted the scaling configuration of apps whose growth was still hung up inside the Admin clusters, which increased their overall latency.

[2022-09-27 21:22 UTC] We didn't see any error spikes in our Admin clusters but kept monitoring it given that our CX told us they were still seeing intermittent errors on Admin modules.

[2022-09-27 22:51 UTC] Our CX team confirmed that they could no longer see errors on our Admin module.

[2022-09-27 22:02 UTC] We declared the incident resolved.

Mitigation strategy

This incident was very complex to coordinate and act on, but we are confident to confirm that our mitigation and follow-up actions are already preventing this failure from happening again.

During this incident, we worked on three mitigation actions:

1. Spin up new clusters to move all the traffic away from unhealthy clusters. This is a complex process that we managed to do in about one hour for two clusters. Firstly, we prioritized moving Stores' traffic to the new clusters, which ended up being the proper approach since we didn't succeed in recovering the network interface from the unhealthy Store clusters.
2. Investigate where in the cluster the problem was and fix it in order to try to recover it. We succeeded in this action on the Admin clusters. But due to the high traffic in the Store clusters, they were unable to recover themselves after the fix.
3. Investigate why the network failure happened and what triggered the failure to avoid this from happening again. We found that a configuration change triggered the network failure, and unfortunately, it wasn't a change that could be detected immediately. With that said, the blast radius was bigger than our change process could prevent because the change was applied in all clusters, and we only saw the latency increase in our clusters after two hours.

Follow-up actions: preventing future failures

We successfully identified the code automation that triggered the network failure and removed it from the change management process to prevent this failure from happening again.

We are already evolving our automation workflow to apply changes to the infrastructure more safely and adding guardrails in the rollout process to decrease the blast radius of a misconfiguration in the infrastructure.

Although we acted when only one cluster was unhealthy, we can improve the cluster's response time alerts to decrease detection time and make us act faster in a future issue with cluster latency.