



PROACTIVE OBSERVABILITY

# Ensuring Cloud Reliability in 2022

What The Outages Of 2021 Taught Us



catchpoint®

# Contents

- 3** Introduction
- 5** Lessons Learned from the Failures of 2021
  - 5** No service is “too big to fail”
  - 5** Any change, manual or automated, can result in system failure
  - 6** Don’t assume only code bugs and infrastructure load cause failures
  - 6** Know the foundations of the network that your code and systems run on
  - 7** Automation is great; but its quality must be the same or better than production systems
  - 8** “Trust and verify” when working with third parties
  - 9** Have a thorough monitoring and observability plan
  - 10** Communicate quickly and clearly to those impacted
  - 11** Practice, practice, practice
- 12** Incident Analysis

# Introduction

2021's slew of Internet outages or disruptions showed how connected and relatively fragile the Internet ecosystem is. Increasingly, we are seeing problems when even a small component of it goes askew.

These days, the likelihood of an issue having a cascade effect have compounded due to:

- The increasing centralization of Internet infrastructure, which results in every outage having a wider footprint.
- Greater complexity: As systems become more complex (due to accelerated digital transformation initiatives), it becomes more difficult to find the origin of an issue and respond quickly.
- The likely permanent shift to hybrid work causes remote workforces to be reliant on distributed cloud-based applications. So now employees are impacted, as well as customers.
- The move to cloud-based applications makes it increasingly difficult to know where issues lie — and frequently, problems lie beyond IT's control. This can leave us at the mercy of third-party networks and providers.

Even for the systems that are within our control, we're all human and can make mistakes that accidentally lead to downtime, no matter how careful we are.



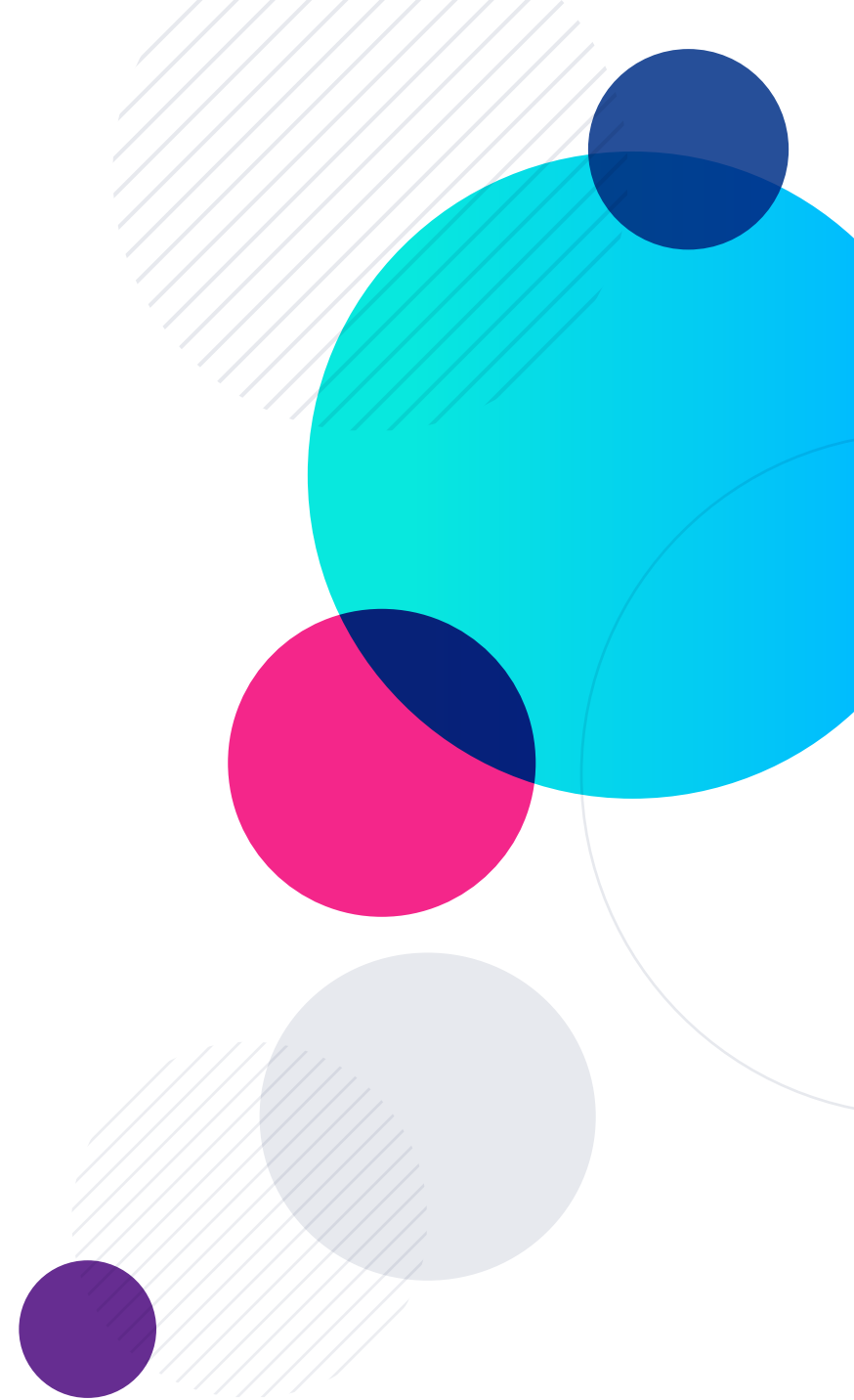
It is obvious that you must implement an incident response process, and what is less obvious, but even more crucial is the fact you need a change management process that nudges teams to better handle changes that can have catastrophic results.

That way, your IT teams and the wider business have a robust plan to follow when an issue occurs (and it will occur, no matter how careful you are). A comprehensive plan should include information that allows you to best understand where problems originate from, exactly what is happening, and what your response to the incident should be.

At Catchpoint, we report on major outages as they happen. We are able to detect them before anyone else, frequently before even the cloud provider updates their status page. Our other unique strength is that we discover the areas impacted, often precisely those that teams do not even think have been impacted.

Every time there is a major incident, we write a blog post teasing apart the details of an issue with the intention of guiding people and businesses on how best to manage these types of incident. We analyze what's going on and why, and what you can do about it (feel free to [subscribe to the blog](#) for quick-response posts). In a collective effort to understand how better to ensure cloud reliability, we have gathered the top pieces of analysis in this eBook.

Let's begin by discussing the top lessons learned and then dive into the details of the incidents themselves.



# Lessons Learned from the Failures of 2021

## No service is “too big to fail”

The biggest lesson of last year is that it does not matter how big a service provider is — they can still fail.

Failures happened from the giants of the Internet: Akamai, Amazon, Facebook, Fastly, and Salesforce. No one can be 100% certain it won't happen to them — or it won't happen again. That's why it's critical that you fully understand what you need to do WHEN (not if) such a failure happens.

*Note: This does not mean outages are acceptable. That said, pretending they don't exist or won't happen is not only pointless, but harmful to your business.*

## Any change, manual or automated, can result in system failure

Almost all outages are caused by a change in code or configuration, which either was done manually by an employee or was the unintended result of automation. At the end of the day, something was changed in the system which resulted in a catastrophic failure.

Obviously, the simple solution to reduce failure is not to make changes. However, equally obviously, that is not what the market expects. As businesses and systems grow, they periodically need to change.

The only solution you're left with is to make sure that for every change, your team knows what to do if a failure happens. This takes rigor, forward thinking, and testing to ensure your teams know what to do and have a plan to execute if change breaks a system.

It is, therefore, imperative to make sure your organization has a proper change management process. This process must also be backed up with solid processes to help manage change and the consequences of change.

For example, every change must be tracked, and every change needs to be tested before being deployed. Most importantly, every time a change is made, you must continuously monitor key services, transactions, and outputs that may be negatively impacted if things go wrong.

At Catchpoint, we require that every scheduled change has a rollback process documented — it might be something simple, but if things go wrong, having it pre-documented is crucial.

## Don't assume only code bugs and infrastructure load cause failures

All of IT understands there are certain things we have under our control and others that are not. Since we tend to perform the most activity around the areas we can control, they're what we pay most attention to. Therefore, we monitor our containers, VMs, hardware, and code. We have unit and automated tests running on our test environments. We have systems that consume all the logs we can afford to consume.

While these are definitely great practices, they are not enough to get in front of outages. Issues can and will occur in other parts of the system — and you can't simply ignore this fact, or you will get into trouble.

Don't just observe the components of the system you have within your control. Observe what you deliver to your consumers or users. Observe their output and performance, even if certain things in the service delivery chain are not within your control, such as third parties like CDNs, managed DNS, and backbone ISPs. While these might not be your code or hardware, your users will still be impacted by any issues they experience, and so will your business.

## Know the foundations of the network that your code and systems run on

We assume that things we haven't changed or modified will continue to work as always, today, and tomorrow. In other words, we think that because we didn't make a change, the system won't break. Hence, we don't pay as much attention to DNS, BGP, TCP configuration, SSL, the networks the data traverses, or any single points of failure in the infrastructure which we rarely change.

This issue is exacerbated by the way in which cloud has abstracted a lot of the underlying network from Dev, Ops and network teams. That can make it harder to find a problem. The result is that when these fundamental components fail, it catches us by surprise. It takes a long time to detect, confirm, or find root cause, because teams were not properly prepared.

Therefore, ensure that you continuously monitor these aspects of your system. Put into place a plan so your teams are trained and know what to do in case of failure. Finally, practice your response, because the longer your team doesn't use a skill (and therefore loses the muscle memory), the longer your outage could last.

Be ready to act: If you are observing your end-to-end experiences, you can act when things that are out of your control are impacting your users. Examples of actions you could take include dropping the third party, switching to a backup solution, or, at the least, communicating to your users what's going on while your teams figure out what to do.

## Automation is great; but its quality must be the same or better than production systems

Automation has been the biggest move of the last decade in IT organization. It helps us be efficient, removes room for error, and simplifies complex tasks. However, we tend not to apply the same rigor to automation as we do to production systems. We can't ignore the importance of designing and testing automation to make sure there are no bugs hiding in the code or script.

For example, look at the Facebook failure from October 2021. A logic problem in the automation led to all the DNS servers being taken out of the BGP announcement, because the servers were not in a proper state.

The result was that there were no DNS servers available, and Facebook couldn't even do the simplest thing any site should during an outage: issue a notice for their users that explains they are down and working on fixing the situation.

Ultimately, this issue was a design flaw in the automation logic. It could have been caught before it was implemented, or at least during testing of the automation script/code. The designer of the system should have thought about what happens if all the DNS servers were impacted... what can automation do?

If you have a general lack of testing and limited awareness of these sorts of issues, you can be taken by surprise when things go wrong. In addition, if you wait until a failure of automation happens to understand the quality of your automation framework and scripts, it will leave too much to do and too much debt collected. Hence, it's best to integrate testing into the automation design and implementation.

### To solve this issue, follow these four simple processes:

- 1 Apply proper design processes, such as documentation and design reviews.
- 2 Ensure you consider edge cases. They are deadly traps.
- 3 Perform functional, automated, and regression testing of automation scripts.
- 4 Test periodically, even if no change to the automation code was made, to ensure everyone knows how it works and what it does.





## “Trust and verify” when working with third parties

When we at Catchpoint are working with companies new to observing digital experience, we often hear, “We are not sure why this AWS outage impacted us since we are not in AWS.” As we help them peel through the layers of their providers, they discover that they indeed have vendors in the critical path of their business transactions who are on AWS and were impacted. Then, they understand that the outage spread like a virus to impact them.

If you have taken courses on managing people or delegation, you probably have heard that one of the key strategies for successfully delegating work is “trust and verify.” This strategy holds true in operating complex systems that rely on multiple teams, and often other vendors like cloud compute, CDN, Managed DNS, etc.

This phrase means that when another team or vendor is making a change, you should not only trust that they did their job in analyzing and planning the change, but also verify that what they have planned is not going to impact you.

For example, if you receive a notice that your key provider is planning a major system update in a week’s time, have your team ready with:

- A crisis call plan (who is on call, what should they do, who should you page if there’s an issue, etc.).
- A plan to mitigate failure from the other party (and make sure that you have tested it ahead of time).
- A clear understanding of what will be impacted by any failure.
- A communication plan and templates that can be easily populated with need-to-know information for users and customers.
- A monitoring and observability plan that covers all the bases.

“An obvious recommendation for websites is that they need to build in more resilience and redundancy when using third-party services. While many of these third-party providers have multiple points of presence, which introduces some redundancy within their infrastructure, they are not immune to failures...

**Moreover, websites need to understand the hidden dependencies of the third-party services they use as they may be indirectly exposed to potential threats. For example, if a website uses multiple CDN providers but those CDNs use the same DNS provider.”**

Analyzing Third-party Service Dependencies in Modern Web Services: Have We Learned from The Mirai-Dyn Incident?



## Have a thorough monitoring and observability plan

A thorough, well-thought-out monitoring and observability plan is essential. This plan must establish baselines for how things looked before the change, so you can compare it with how things look after.

Some examples of why this is important include:

- If you're updating your firewall rules or network device firmware, are you sure that it hasn't started dropping some connections, or added latency to every packet sent?
- If your DNS vendor has scheduled a maintenance window to upgrade their systems, do you know how long a DNS lookup took before, so you can make sure it hasn't gotten slower?
- If your engineering team is pushing an updated version of your application, do you have all the relevant baselines measured? Speed? Size? Responsiveness? What else is important to your users?
- Do you have data on how time of day, day of week, holidays, or special events change the performance of the application/service?
- Are you monitoring services from both inside the production environment (i.e., inside your firewall) and outside to ensure 360-degree visibility into the experience for both your internal systems and the external world?
- Are you looking only at code tracing, logs, and CPU, or also continuously testing what matters the most: the output of the service from the perspective of the consumer (user, machine, etc.)?



## Communicate quickly and clearly to those impacted

When some businesses go down, only their direct customers get impacted. However, for others, it's a different story. The impact of their outage is a ripple effect, which affects the customers of their customers.

The Internet has created a complex web of relationships between companies providing a variety of aspects of a user's experience. When a CDN goes down, for instance, it might impact an analytics company, which, in turn, might impact web publishers who were not even using the CDN. This domino effect causes a lot of noise and stress on your customers, as well as everyone else who is impacted.

Therefore, communicating clearly to your direct customers, and to the entire world, is critical. Ensure you have a proper process around this type of communication, templates of communication, and a clear communication cadence on multiple fronts.

In addition, if there is an impact on a large population of the Internet, make sure you have a public relations plan and ensure your PR team is ready to take control of media communications. If you are making the news, it is always better that you "own" the story. Otherwise, you leave your business open to rumors, negative commentary, and speculations on Twitter, Reddit, and major news outlets — and from your competitors.



## Practice, practice, practice

The biggest lesson might be that businesses need to ensure their teams practice what to do when failure happens. What we at Catchpoint observed in most of the outages of 2021 is that a lot of teams were not prepared. It took too long to identify issues and figure out what to do, and the mean time to resolve suffered.

You need to ensure your teams are ready for failure. Business systems have grown in complexity. Single points of failure are spread out across that complexity. Organizations rely on external providers more than ever and those provider systems have often become even more complex than our own.

Implement the following steps, so you'll be prepared for failures:

- Put a monitoring and observability strategy in place. This is what will detect and let your team react at speed when an “outage hits the fan.”
- Design and test a crisis process.
- Develop robust playbooks or run books.
- Plan for the outages of key vendors.
- When practicing a crisis, turn it into a game — but practice, so people are not struggling when that outage occurs.

“The SAP leadership is interested in two kinds of situations when it comes to monitoring: the first is to avoid any kind of doubt, and the second is if downtime happens, to be as fast and as close to the customer as possible.

**Catchpoint is really our main tool for outage detection. Once an alert in Catchpoint is received, we have a team that analyzes the request. If the analysis shows we have a business down situation, the availability detection & notification team starts the notification process and informs our situation room.**

**We actually trigger Catchpoint tests to verify if the site is behaving as it should. We get Catchpoint alerts within seconds when a site is down. Within three minutes, we can identify exactly where the issue is coming from, inform our customers and work with them.”**

Martin Norato Auer, VP of CX Observability Services,  
SAP Commerce Cloud

# Incident Analysis

## Contents

<b>Incident Reviews: December 7, December 15 and December 22, 2021 - What Can We Learn from AWS' December Outagepalooza? .....</b>	<b>15</b>
1. Early detection is key to handling outages like the AWS incidents.....	15
2. Comprehensive observability helps your team react at speed to outages. ....	17
3. Ensuring your company's availability and business continuity is not a solo endeavor.....	17
4. Depending on only a monitoring solution hosted within the environment being monitored is not enough.....	17
<b>Incident Review: November 16, 2021 - Google Cloud Outage has Widespread Downstream Impact... 19</b>	<b>19</b>
In the shoes of the end user: the broken Google bot .....	19
The latest outage of 2021 .....	20
Catchpoint saw a sudden burst of test failures .....	20
"A latent bug in a network configuration service" .....	21
Three key lessons for any company .....	22
<b>Incident Review: October 7, 2021 - The Ripple Effect of A BGP Misconfiguration at Telia .....</b>	<b>23</b>
What we saw at Catchpoint .....	23
The mystery is solved: BGP misconfiguration .....	24
Let's return to the time of the cri(me)sis .....	25
What can you do to tackle BGP misconfigurations? .....	26
<b>Incident Review: October 4, 2021 - A Case of Social Networks Going Anti-social at Facebook .....</b>	<b>27</b>
Facebook is everywhere... it's beyond just social media .....	27
Alarms started at Catchpoint when we detected server failures.....	28
A tale of badge failure and BGP .....	32
A deep dive into the BGP data .....	33
Having a quick response is key, as long as your badge is working! .....	34
Update from Facebook's postmortem analysis .....	34
<b>Incident Review: September 29-30, 2021 - Issues Caused by the Let's Encrypt DST Root CA X3 Expiration.....</b>	<b>35</b>
Do you trust Let's Encrypt? .....	35
How digital certificate trust works .....	36
R3 certificate expiry and the chain of trust.....	37
September 30, 14:00 UTC: DST root CAx3 certificate expiry and its consequences .....	38
Fixes, fixes... ..	40
Fixing this on your server .....	41
In conclusion.....	41

<b>Incident Review: October 1, 2021 - An Account of DNS Misconfiguration at Slack .....</b>	<b>42</b>
Slack acknowledges the issue .....	42
Why monitoring from the cloud isn't enough .....	43
Catchpoint's last mile tests detected DNS issues as the root cause .....	44
Understand how to resolve DNS issues more quickly .....	45
<b>Incident Review - September 7, 2021: A Case of Dr. BGP Hijack or Mr. BGP Mistake at Spectrum? .....</b>	<b>46</b>
Fact checking the Spectrum investigation .....	47
So... was it Dr. BGP hijack or Mr. BGP mistake? .....	48
Lessons for network administrators .....	50
<b>Incident Review: August 31, 2021 - AWS Services Hit by Major Spikes in Response Times .....</b>	<b>51</b>
What the AWS status dashboard showed .....	51
Root cause identified: network connectivity issues .....	52
Catchpoint detects and alerts on AWS outages first .....	54
The impact of AWS' outage on active observability services running on AWS .....	55
How can you prevent single points of failure? .....	56
<b>Incident Review: August 31, 2021 - High CDN Network Response Times Slow Down Major Websites Worldwide .....</b>	<b>57</b>
Performance issues create problems at Akamai .....	57
The four pillars of Digital Experience Management .....	57
The three network components involved in using a CDN .....	57
Major websites hit by high response times .....	58
Availability drops due to 503 error codes .....	60
Potential causes of latency in CDN networks .....	60
Be prepared to act quickly .....	61
<b>Incident Review: June 8, 2021 - Fastly Outage Impacts Major Websites Worldwide .....</b>	<b>62</b>
Internet outage timeline .....	62
Fastly acknowledges the problem .....	65
Why employ a multi-CDN strategy? .....	67
Conclusions and advice .....	67
<b>Incident Review: May 11, 2021 - Anatomy of The Salesforce Outage .....</b>	<b>68</b>
"We drink our own champagne at Catchpoint" .....	69
Our monitoring strategy for Salesforce .....	69
The Salesforce summary of the outage .....	74
Developing a sound monitoring and observability strategy for SaaS .....	74

**Incident Review: May 6, 2021 - How Catchpoint Resolved Issues Caused by the Neustar UltraDNS**

<b>Outage .....</b>	<b>75</b>
What happened with UltraDNS? .....	75
What did the velocity and volume of the tickets that were coming in look like? .....	76
What did clients experience? .....	76
How we ensure support relationships are true partnerships .....	77
An ideal customer support response to an Internet outage .....	78
<b>Conclusion.....</b>	<b>79</b>
Stay up to date on the most recent outages .....	79
About Catchpoint .....	79

# Incident Reviews: December 7, December 15 and December 22, 2021 – What Can We Learn from AWS' December Outagepalooza?

By Carol Hildebrand and Raj Jathar

2021's slew of Internet outages or disruptions show how connected and relatively fragile the Internet ecosystem is. Case in point: December's trifecta of Amazon Web Services (AWS) outages, which really brought home the fact that no service is too big to fail:

**12/07/2021.** Millions of users were affected by this extended outage originating in the US-EAST-1 region, which took down major online services such as Amazon, Amazon Prime, Amazon Alexa, Venmo, Disney+, Instacart, Roku, Kindle, and multiple online gaming sites. The outage also took down the apps that power warehouse, delivery, and Amazon Flex workers—in prime holiday shopping season. The AWS status dashboard noted that the root cause of the outage was an impairment of several network devices.

**12/15/2021.** Originating in the US-West-2 region in Oregon and US-West-1 in Northern California, this incident lasted about an hour and brought down major services such as Auth0, Duo, Okta, DoorDash, Disney, the PlayStation Network, Slack, Netflix, Snapchat, and Zoom. According to the AWS status dashboard, "The issue was caused by network congestion between parts of the AWS Backbone and a subset of Internet Service Providers, which was triggered by AWS traffic engineering, executed in response to congestion outside of our network."

**12/22/2021.** This incident was triggered by a data center power outage in the U.S.-EAST-1 Region, causing a cascade of issues for AWS customers such as Slack, Udemy, Twilio, Okta, Imgur, Jobvite and even the NY Court system web site. Although the outage itself was relatively brief, related effects proved vexingly persistent, as some AWS users continued to experience problems related to the issue up to 17 hours later.

The reality is, the next outage is not if, but when, where, and for how long. Pretending they don't exist or won't happen is not only pointless but harmful to your business. Looking back at the three December outages, we see four key takeaways:

## 1. Early detection is key to handling outages like the AWS incidents.

Catchpoint observed all three outages well before they hit the AWS status page:

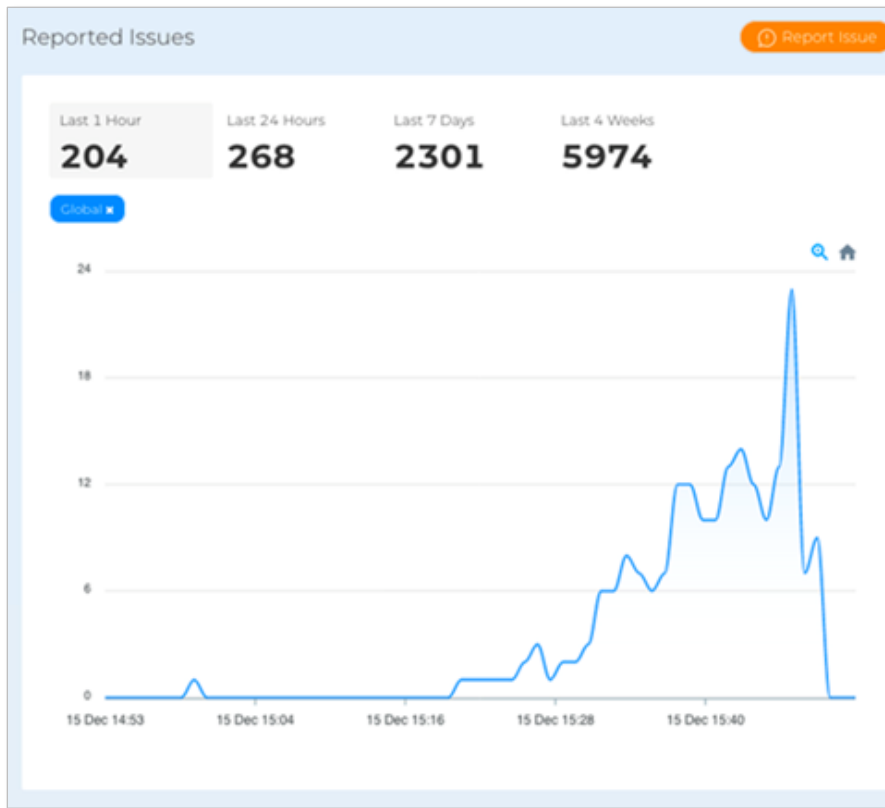
**12/7/2021:** Here at Catchpoint, we observed connectivity issues for AWS servers starting at 10:33 AM ET, considerably earlier than the announcement posted to the AWS Service Health Dashboard at 12:37 PM EST.



Waterfall graph showing 504 error response for HTML page of Amazon site (Catchpoint)

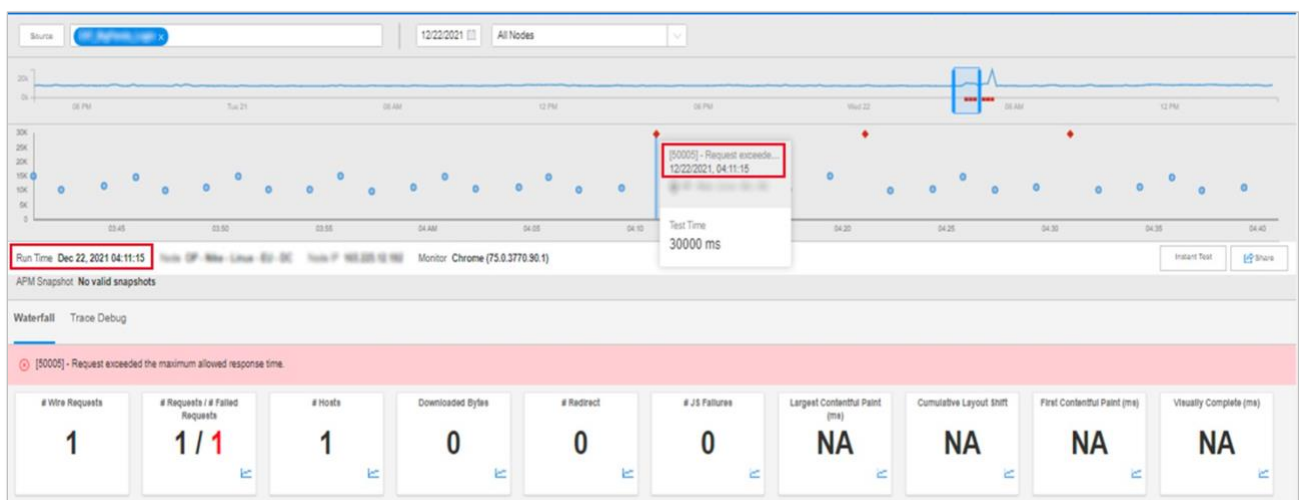


12/15/2021: Catchpoint noted the outage at approximately 10:15 AM ET, once again before the AWS announcement at about 10:43 AM ET.



User sentiment analysis (Catchpoint)

12/22/2021: Catchpoint first observed issues at 07:11 AM ET, 24 minutes ahead of the AWS announcement.



Proactive Chrome browser observer showing AWS outage (Catchpoint)

Early detection allows companies to fix problems potentially before they impact customers and implement contingency plans to ensure smooth failover as soon as possible. If the issue continues, it also allows them to proactively inform customers with precise details about the situation and assure them that their teams are working on it.

## **2. Comprehensive observability helps your team react at speed to outages.**

While it may be tempting to leave AWS monitoring to, well, AWS, that could leave you in the dark, observability-wise. A comprehensive digital observability plan should include not only your own technical elements, but also service delivery chain components that are not within your control. For example, you need insight into the systems of third-party vendors such as content delivery networks (CDNs), managed DNS providers, and backbone Internet service providers (ISPs).

While these might not be your code or hardware, your users will still be impacted by any issues they experience, and so will your business. If you are observing your end-to-end experiences, you can act when things that are out of your control are impacting your users.

It also means continuous observation of your systems to detect failure of fundamental components such as DNS, BGP, TCP configuration, SSL, the networks the data traverses, or any single point of failure in the infrastructure that we rarely change.

This issue is exacerbated by the fact that cloud has abstracted a lot of the underlying network from development, operations, and network teams. That can make it harder to find a problem.

As a result, it can catch us by surprise when these fundamental components fail. If teams are not properly prepared, it adds needless – and costly – time to detect, confirm, or find root cause. Therefore, ensure that you continuously monitor these aspects of your system and train your teams on what to do in case of failure.

## **3. Ensuring your company's availability and business continuity is not a solo endeavor.**

The AWS incidents all clearly illustrate the downstream effect that an outage at one company can have on others. Digital infrastructure will assuredly continue to grow more complex and interconnected. Enterprises today run systems that run across multiple clouds. They also rely on multiple teams, often including a raft of other vendors, such as cloud compute, CDNs, and managed DNS.

When issues originating with outside entities such as partners and third-party providers can bring down your systems, it is time to build a collaborative strategy designed to support your extended digital infrastructure. For that, comprehensive observability into every service provider involved in the delivery of your content is crucial.

## **4. Depending on only a monitoring solution hosted within the environment being monitored is not enough.**

While there are many monitoring solutions out there, make sure that you have a "break glass system" to be able to failover to a solution outside of the environment being monitored. Many visibility solutions are located in the cloud, which makes them vulnerable when cloud technologies go down.

This is why ThousandEyes, Datadog, Splunk (SignalFX), and NewRelic all reported impacts from the 12/07/21 and 12/15/21 events.

During the first event, [Datadog](#) reported delays that impacted multiple products, [Splunk \(SignalFX\)](#) reported that their AWS cloud metric syncer data ingestion was impacted, and [NewRelic](#) reported that some AWS Infrastructure and polling metrics were delayed in the U.S.

There were also a number of issues triggered by the [12/15/2021 event](#):

- Datadog reported delays in collecting AWS integration metrics.
- ThousandEyes reported degradation to API services.
- New Relic reported that their synthetics user interface was impacted, as well as some of the APM alerting and the data ingestion for their infrastructure metrics.
- Dynatrace reported that some of their components hosted on AWS cluster were impacted.
- Splunk (Rigor and SignalFX) reported increased error rates in the West coast of the U.S. and a degradation in the performance of their Log Observer.

Lack of observability is never a good thing, but over the course of an outage, it is significantly worse.

---

*"The first thing that would be useful is to have a monitoring system that has failure modes which are uncorrelated with the infrastructure it is monitoring."*

~[Adrian Cockcroft, VP, Amazon Sustainability Architecture, AWS](#)

---

*Published on Feb 15, 2022*

## Incident Review: November 16, 2021 – Google Cloud Outage has Widespread Downstream Impact

By Dritan Suljoti

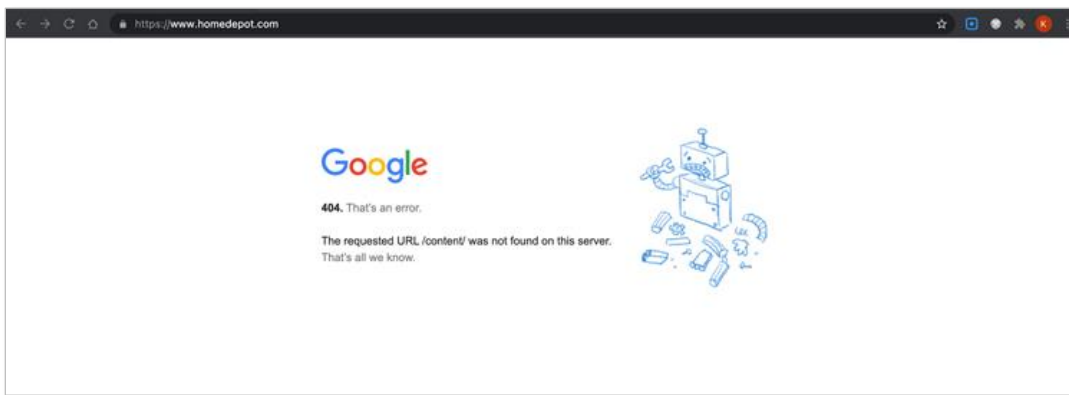
Outages on the Internet always catch you by surprise, whether you are the end user or the Head of SRE or DevOps trying to keep a clear mind while you execute your incident playbook.

As people in charge of ensuring reliable services for our customers, our normal experience of outages involves surfing a deluge of fire alarms and crisis calls as we work to solve the problem as quickly as we can. We often forget, therefore, what an outage means to the end user.

On Tuesday, November 16, 2021, however, I was reminded of exactly how the shoe feels on the other foot.

### In the shoes of the end user: the broken Google bot

On Tuesday, as I was trying to purchase something for my home on Homedepot.com, my browser rendered an unusual page: the Google bot page with its 404 message.



Google 404 Error Page (Google)

Surprised, I clicked “reload.” Nope, still the same page. I typed the URL again with www. and then without it, but still the same broken Google bot greeted me.

“Well, there’s no way Google acquired Home Depot,” I thought (although its parent company is missing a company that starts with the letter H).

Being a tech geek, the next thing in my mind was maybe I was somehow using Google’s Public DNS 8.8.8.8 and the DNS lookup was failing and Google had decided to launch a new feature that routed non-resolvable domain to their IP (a technique we’ve seen a few companies use before)? But no, neither of those was it either.

Giving up on being a consumer and blind to what was incomprehensible, I simply logged onto the Catchpoint platform to see what was going on. The answer was immediately clear: multiple sites failing, all experiencing the same error message and all of them customers of Google Cloud. I visited Google's status page, and nothing was posted yet... and there was still nothing on it for another thirty minutes from when the problems started.

Let's take a quick look at the incident itself.

## The latest outage of 2021

Tuesday November 16, starting soon after midday ET, many companies not owned by Google saw their websites knocked offline, replaced by the Google 404 page.

What was going on?

Google didn't acquire your favorite site then shut it down. In fact, it was collateral damage due to the latest outage of 2021, this time on Google Cloud, which many, many companies rely on for hosting. The impact on a lot of these companies would have been lost revenue and possible damage to business reputation.

## Catchpoint saw a sudden burst of test failures

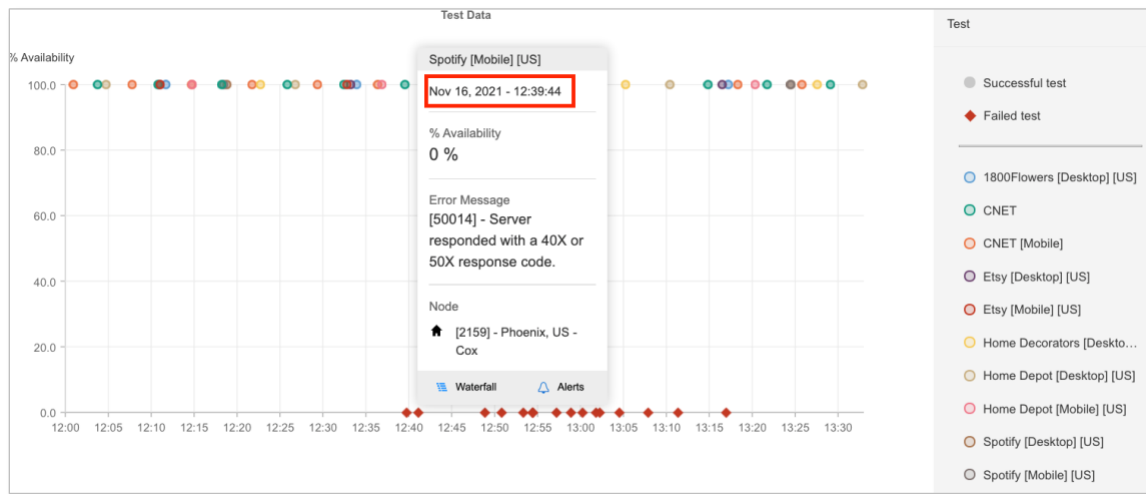
At Catchpoint, we saw a sudden burst of test failures, beginning at 12:39pm ET. It impacted many companies, large and small. Some of the businesses that were affected include the likes of Nest, 1800Flowers, CNET, Home Depot, Etsy, Priceline, Spotify, and Google itself.

By around 13:10, the problem was partially resolved. However, [according to Google](#), the issue did not get fully resolved for all impacted products for almost two hours, lasting until 14:28 am ET. Some companies were back online quickly, while others continued to experience errors or long loading times for some time.

Below is a chart showing availability of many of these sites. You can easily see where the sudden plunge from the cliff edge took place, diving from steady high availability to 0%.



Failed tests show the impact on many of our customers (Catchpoint)



Spotify dashboard showing availability at 0% (Catchpoint)

The customer impact varied according to the Google Cloud service it depended on. For instance, Google App Engine saw an 80% decrease in traffic in central parts of the U.S. and portions of Western Europe. Google Cloud Networking customers were unable to make changes to website load balancing, which led to the 404 error pages.

Indeed, it was not just web pages that were impacted, but multiple Google Cloud products, including:

- Google Cloud Networking
- Google Cloud Functions
- Google Cloud Run
- Google App Engine
- Google App Engine Flex
- Apigee
- Firebase

### **“A latent bug in a network configuration service”**

Google Cloud apologized for the service outage and any inconvenience it caused to its downstream customers. On its [status page](#), the organization specified root cause as “a latent bug in a network configuration service which was triggered during a leader election change.” The cloud giant has assured customers there are now “two forms of safeguards protecting against the issue happening in the future.”

With the massive adoption of public cloud, this latest incident (in a long year of outages) illustrates how significant the impact a public cloud vendor outage can be downstream. It also illustrates how vulnerable enterprises are to third-party vendor outages.

You can clearly see that many enterprises rely on public Internet, services, and infrastructure to conduct their business and deliver digital experiences to their clients. While there are many positives to this situation, the challenge is that those same businesses have little to no control over the underlying infrastructure on which their organizations run.

## Three key lessons for any company

Below are three critical lessons that we took away from this outage, which you can apply to your own business:

### LESSON 1

While failure is bound to happen, don't overlook the importance of communicating to the end user through a proper error page.

Don't assume your users will find your status page or go to Twitter to see your communications. They will have already moved on to your next competitor and will eventually read about it in the news.

If there was one thing that confused the end users in this instance, it was the Google error page that greeted people. Most people would have expected an error message from the company they were trying to reach, not the hosting company. It's a little unclear if Google Cloud allows folks to modify this. Perhaps it's not possible.

However, any company should be ready for such failures, and implement a process where they are able to change the DNS or CDN configuration to point people to a proper error page with their own branding and messaging to apologize for the failure in their own words. And ideally, make it fun. Don't be afraid to be human and relate to the end users. A proper error page is always better than confusing error pages (as in this instance), obscure errors (such as "the server failed to respond"), or worse, nothing at all and hanging on into infinity to connect to the server.

### LESSON 2

Ensure you implement proper observability of your services, which means from outside your firewall, datacenter, or cloud.

While many observability platforms have defined "observability" to fit their products (tracing and logging) - [in reality observability had its origins long before tracing came about](#). In control theory, observability is defined as a measure of how well the internal states of a system can be inferred from the knowledge of its external outputs.

This won't have been the first time that a company will have learned they are down from the news or a customer complaining. You do not want to find out about the problem in this way. Far better to stay ahead of it by observing your services from outside your cloud provider. When you are relying on code tracing and logs alone, you won't see the problem.

Be proactive and stay on top of your services, and the services and infrastructure providers you rely on that are single points of failure.

### LESSON 3

Finally, track the SLA of your services, and know your MTTR.

You need to track how good your teams and providers are at resolving issues. This is how you build trust and verify people are doing what they are accountable for.

Real time data from an independent monitoring and observability solution will allow you to find out precisely when the issue started and when it was resolved. You cannot rely on status pages to be accurate about the impact the problem had on your site. Everyone will have been impacted differently: slightly earlier or later, shorter, or longer...

*[Published on Nov 18, 2021](#)*



## Incident Review: October 7, 2021 - The Ripple Effect of A BGP Misconfiguration at Telia

By Alessandro Improta, Anna Jones, and Luca Sani

On Thursday October 7th, Telia, a major backbone carrier in Europe, suffered from a network routing issue between 16:00 and 17:15 UTC. This had a significant ripple effect with several other major companies reporting outages at around the same time.

Companies affected included:

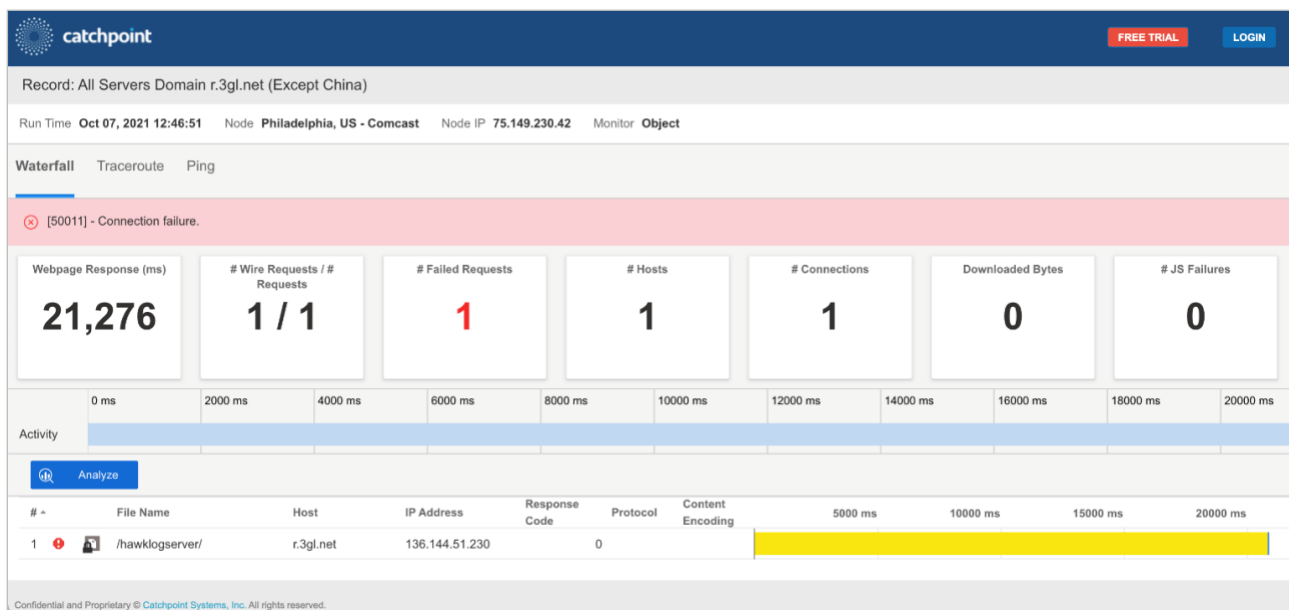
- [Cloudflare](#)
- [Equinix Metal](#)
- [Fastly](#)
- [Firebase](#)
- [NS1](#)

It's always startling to see the secondary and tertiary effects that a major outage can have. In this instance, it briefly caused some of the world's biggest infrastructure and content delivery networks to have serious performance issues in the U.S., Canada, and 12 other countries.

There is a common theme in the slew of outages 2021 has brought us: the added - and painful - impact that an outage in one part of the Internet delivery chain can have on all third-party dependencies.

### What we saw at Catchpoint

Many of our customers saw huge increases in webpage response times at the time of the Telia incident. We saw increases in response time across portions of the U.S. East Coast and Europe.



Waterfall showing STCP connection timeout of request to Equinix Metal. (Catchpoint)

## The mystery is solved: BGP misconfiguration

What caused the outage to occur? Let's break down the incident and figure it out.

At 21:17 UTC, Telia Carrier posted the following statement:



Outage Tweet from Telia. (Twitter/@TeliaCarrier)

Just as with previous [Facebook](#) and [Spectrum](#) outages, there was a common villain: BGP. In this instance, the outage was directly caused by a BGP misconfiguration.

---

*"An engineer in another department of a large company may change their own process for the better after reading about an incident written by someone in an unrelated department that didn't directly impact them at the time. This is where distribution comes into play. At the extreme end of this, which I'm hoping we're trending towards as an industry overall, is making postmortems public to get the maximum downstream learning impacts across the entire industry and not just within a single company."*

~John Egan, Former co-founder and Product Lead Workplace, Facebook

---

Ultimately, only a postmortem from Telia could shed light exactly on what happened. However, from digging into the BGP data available, we uncovered some remarkably interesting facts...

## Let's return to the time of the cri(me)sis

Earlier in the evening, Telia shared [this email with their customers](#) (which we accessed from the outage.org mailing list). This note included details of the root cause:

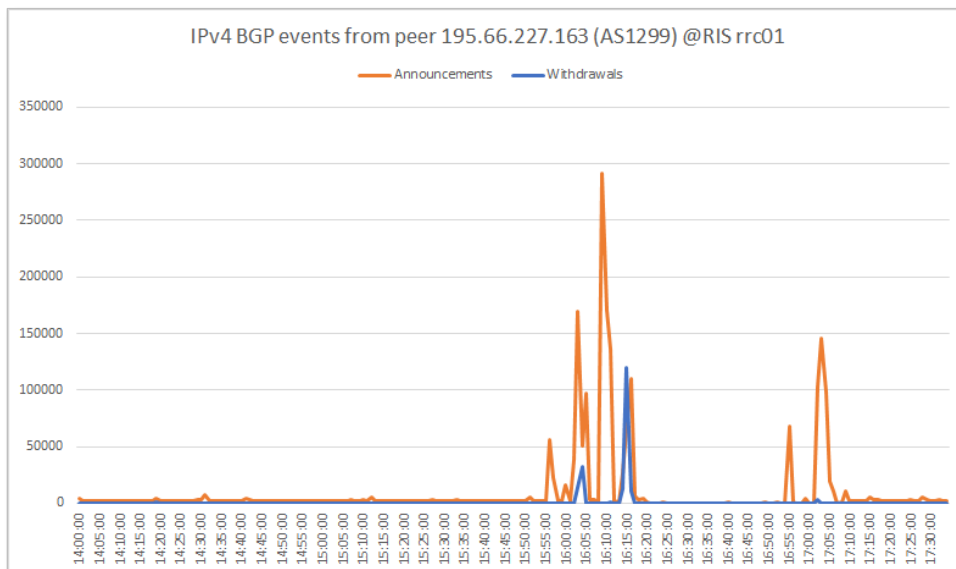
"Dear Customer,

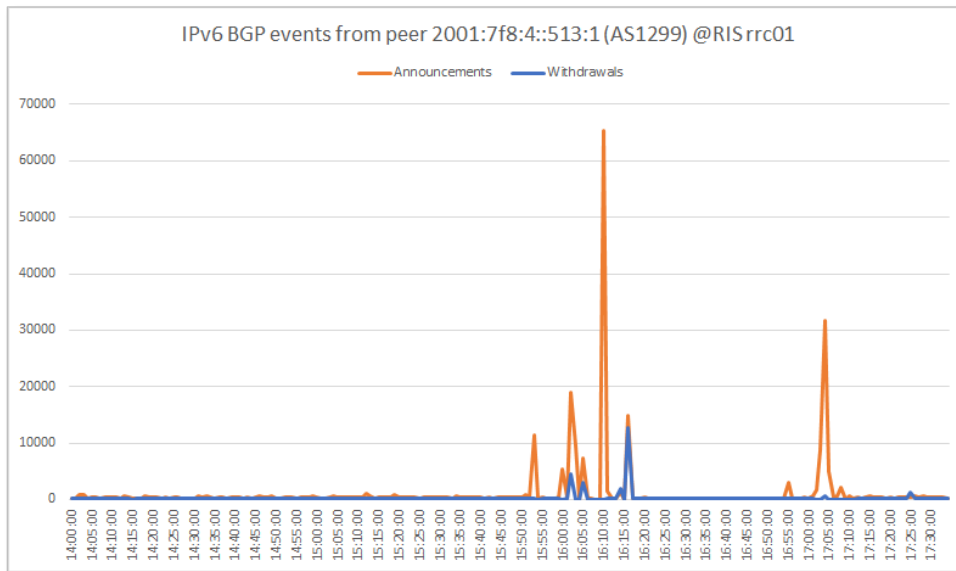
We regret to inform you that your services were affected by an incident that occurred at 16:00 UTC during a routine update of a routing policy for aggregated prefixes in the Telia Carrier IP Core network. This caused traffic to prefixes contained within the aggregates to be blackholed, resulting in an impact on some parts of the network.

When the underlying problem source was traced, the configuration was rolled back to the earlier working version of the routing policy (17:05 UTC). Affected services started to recover gradually after this operation was applied."

The times specified are exactly when we saw the first batch of BGP events.

We focused our attention on rrc01, the RIS route collector that RIPE NCC deployed at the London Internet Exchange (LINX). This router collects data directly from an IPv4 and an IPv6 peer from Telia (AS1299). As can be seen from the following graphs, at around 16:00 UTC, the number of networks announced and withdrawn from the two peers spiked upwards across both peers.





Data collected from IPv4 and IPv6 peers showing withdrawn networks. (London Internet Exchange)

It is interesting to note that the two peers of AS1299 generated BGP events related to about 500k IPv4 networks and 32k IPv6 networks. In other words, more than 50% of the full set of IPv4 routes were affected, as were more than 30% of the full IPv6 routes shared by the peers. This gives us a rough initial idea of how widely the outage at Telia affected the entire Internet.

### What can you do to tackle BGP misconfigurations?

It's the duty of every network operator to avoid misconfigurations in the router they manage. However, no one is perfect and BGP (misconfigurations) happens! That's why businesses need a strong observability and monitoring structure in place to alert them as soon as anything like this begins to spread in the wild. It is critical to react swiftly to the issue, to minimize the disservice for end users.

It's easy to see from the BGP data that, as soon as the BGP instability at Telia started, several operators decided to temporarily switch off their peering with AS1299 and/or attempted to route the traffic on alternative routes.

While it is impossible to avoid BGP misconfiguration completely, network operators should follow common sense rules and apply some of the best practices advocated by [MANRS](#). This will help enable them to minimize the chances of a BGP misconfiguration.

*Published on Oct 8, 2021*

## Incident Review: October 4, 2021 – A Case of Social Networks Going Anti-social at Facebook

By Zachary Henderson, Alessandro Improta, and Anna Jones

In a highly unusual state of events, Facebook, Instagram, WhatsApp, Messenger, and Oculus VR were [down simultaneously around the world](#) for an extended period, Monday, October 4th.

The social network and some of its key apps started to display error messages before 16:00 UTC. They were down until approximately 21:05 UTC, when things gradually began to return to normal.



Facebook Tweet acknowledging the outage. (Twitter/@Facebook)

Can humanity survive hours without the most important social media conglomerate of our time? On a more serious note, as [some users pointed out on Twitter](#), did the global outage highlight the challenges of such a dominant single technological point of failure?

### Facebook is everywhere... it's beyond just social media

What quickly became clear to us at Catchpoint was the fact that the outage was impacting the page load time of many popular websites that are not powered by Facebook. Why? Because Facebook ads and marketing tags are on almost every major website.

Here's an aggregate of the 95th percentile of onload event time, referred to as document complete, alongside the availability and Catchpoint "bottleneck time" impact metric of the site's embedded Facebook content, across the IR Top 100 sites, as measured from Catchpoint's external active (synthetic) observability vantage points.



Metrics showing significantly higher page load times for Facebook users. (Catchpoint)

Note the measurement of *document complete* spikes and sustains at 20+ second higher at 15:40 UTC. These indicate that overall page load times for users were much higher than normal.

*"People and businesses around the world rely on us every day to stay connected. We understand the impact that outages like these have on people's lives, as well as our responsibility to keep people informed about disruptions to our services. We apologize to all those affected, and we're working to understand more about what happened today so we can continue to make our infrastructure more resilient."*

~Santosh Janardhan, VP, Infrastructure, Facebook

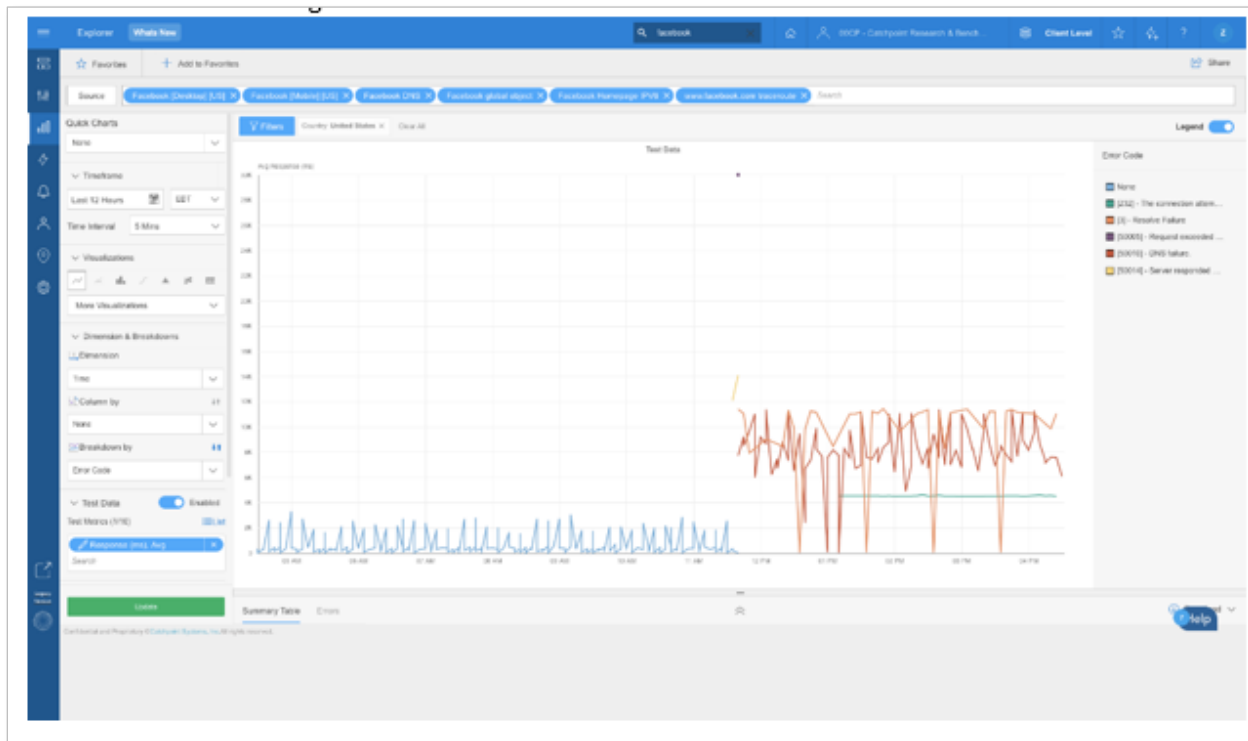
## Alarms started at Catchpoint when we detected server failures

Here at Catchpoint, alarms started to trigger around 15:40 UTC. These alarms resulted from the fact that some of our HTTP tests for Facebook, WhatsApp, Instagram, and Oculus domains started to return HTTP error 503 (service unavailable).

It's worth noting that we do this type of monitoring as part of a benchmarking process. In this way, we can provide insights into the Internet as a whole.

We usually see that Facebook is a highly stable system. The business has built a scalable, reliable, global service. Therefore, when we saw alarms about a Facebook outage, it was easy to determine this was a significant problem.

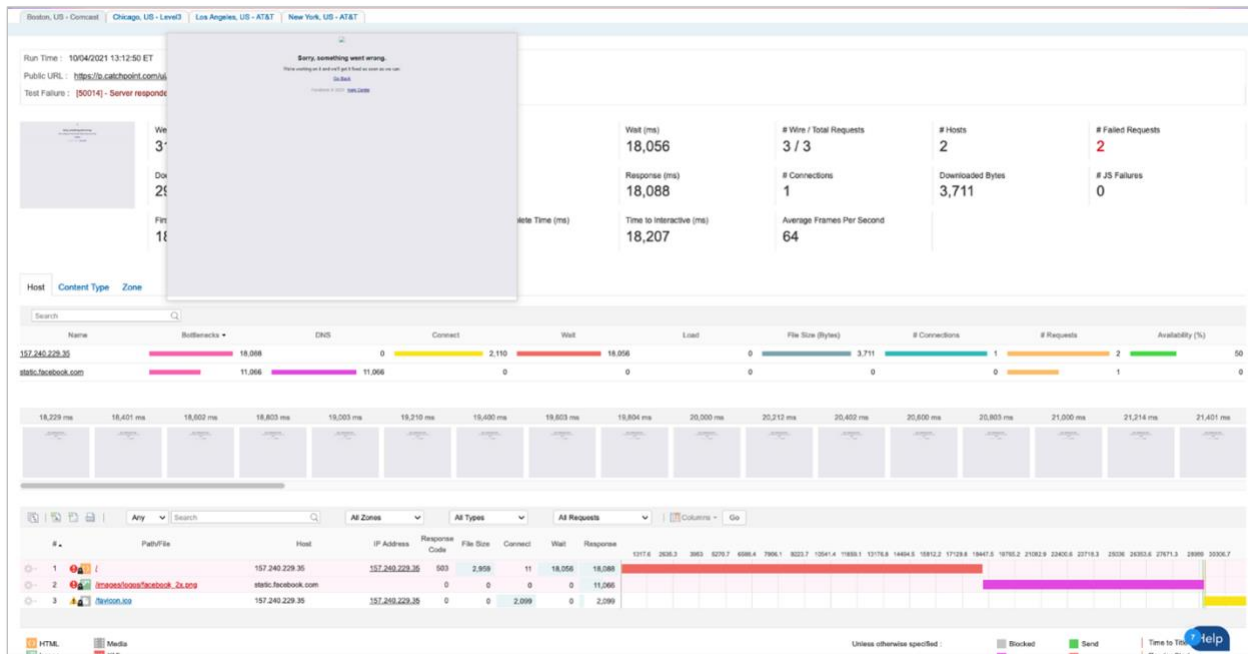
The snapshot below is from the Catchpoint Data Explorer. It shows the server failures that first alerted us to the outage.



Data Explorer snapshot showing Facebook server failures. (Catchpoint)

Five minutes later, we saw that the TTL of the DNS records of Facebook had expired. Shortly after, it became clear that no Facebook nameserver was available, and every DNS query towards `www.facebook.com` was resulting in a SERVFAIL error (meaning a DNS query failed because an answer cannot be given).





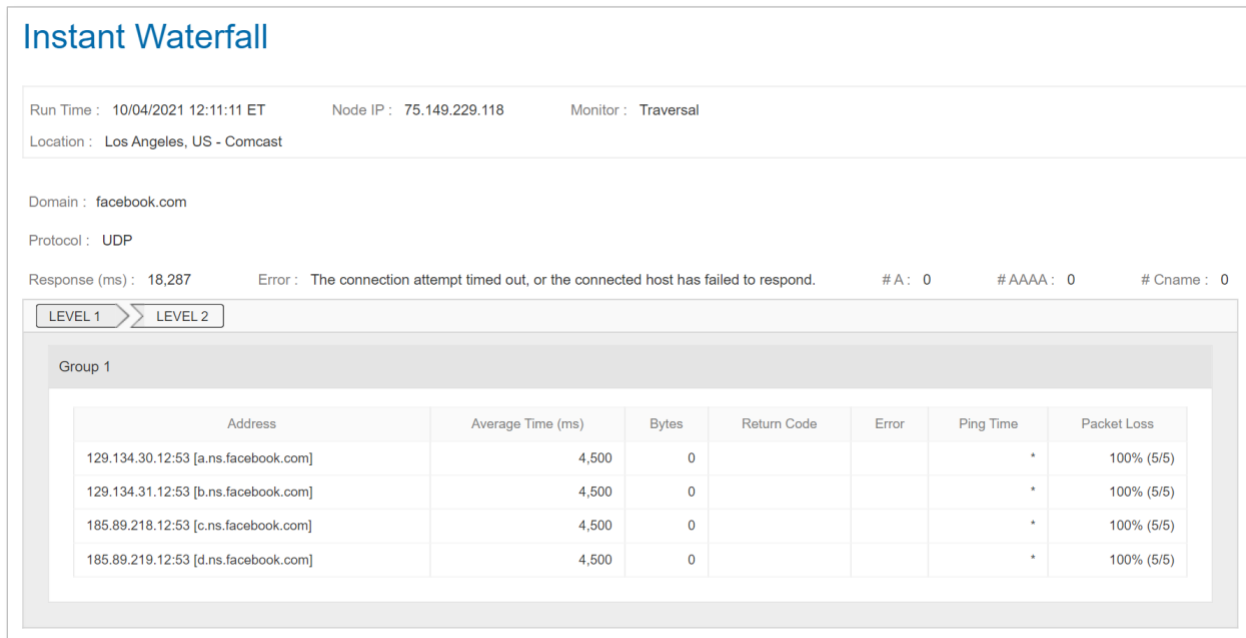
Facebook DNS records showing SERVFAIL errors. (Catchpoint)

Below are examples of the types of HTTP headers 503 errors seen initially:

- HTTP/2 503
- access-control-allow-origin: \*
- content-length: 2959
- content-type: text/html;
- charset=utf-8date: Mon, 04 Oct 2021 16:48:36 GMT
- proxy-status: no\_server\_available;

You can see that it was initially returning a server failure. When DNS records were cached, Facebook's edge was unable to find an upstream proxy server as part of their communication setup.

The next set of screenshots show that when we queried Facebook's top-level domain servers, they weren't working.



Instant Waterfall showing the top-level Facebook domain servers are down. (Catchpoint)

Everything up to now led us to think that the cause of the issue was DNS... But was it?

---

*"Facebook is the Internet in a lot of communities in Southeast Asia," says. "Use of the Internet is really just Facebook and WhatsApp."*

~Ross Tapsell, Senior Lecturer, Australian National University, Asia and the Pacific

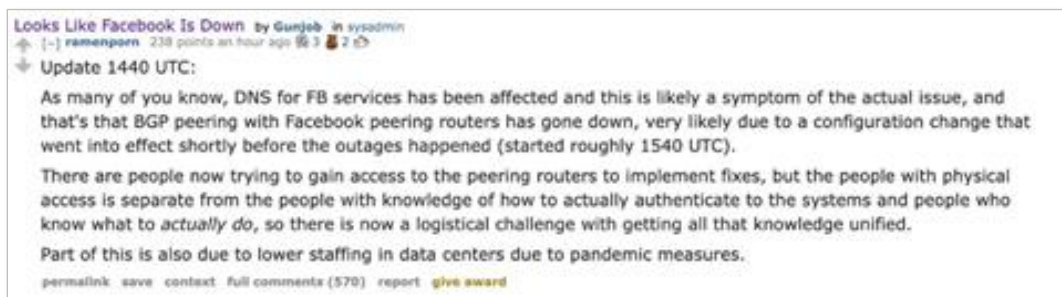
---

## A tale of badge failure and BGP

A lot of speculation around the incident occurred on the surviving social media platforms.



Tweet sharing rumor that Facebook door badges were not working. (Tweet/Sheera Frenkel)

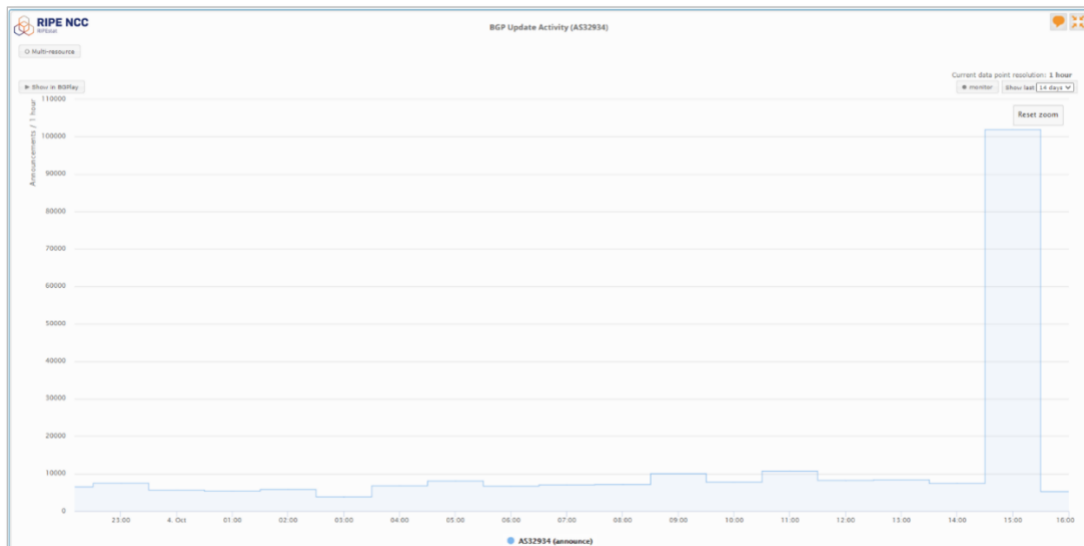


Reddit comment about BGP peering involved in Facebook incident (Reddit/uGunjob)

We may never know if the [Facebook technical staff were indeed locked out of the server room](#) and unable to fix their routers. At the same time, there is some truth to this final speculation: BGP was, indeed, heavily involved in this incident.

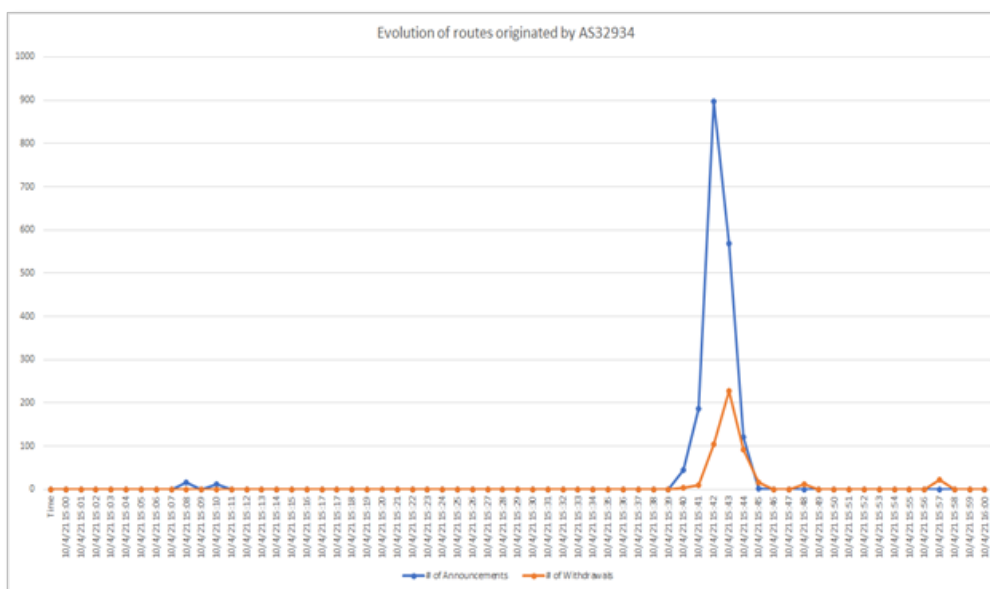
## A deep dive into the BGP data

Facebook manages AS 32934. The networks it originates are usually stable, as can be seen from RIPEstat ([RIPEstat - Ui2013/AS32934](#)).



RIPE NCC data showing a spike in the number of BGP events. (RIPEstat)

Something changed, however, at around 15:40 UTC. At that time, you could clearly see a spike in the number of BGP events. We focused on BGP data collected by RIS rrc10 collector deployed at the Milan Internet Exchange (MIX) between 15:00 UTC and 16:00 UTC.



Evolution of routes originated by AS32934. (Milan Internet Exchange)

From a quick look at the snapshot of 08:00 UTC, AS 32934 was originating 133 IPv4 networks and 216 IPv6 networks. The update messages made it easy to spot that Facebook withdrew the routes to reach eight of those IPv4 networks and fourteen of those IPv6 networks around 15:40 UTC. This was exactly the time when all the Catchpoint alerts started to trigger, and people began to complain about outages.

Even though just a handful of networks experienced outages, this incident demonstrates that it's not the quantity of networks that matters. Some of the withdrawn routes were related to the Authoritative DNS nameservers of Facebook, which couldn't be reached any more. This led to DNS resolutions from all over the world failing. Eventually, it resulted in DNS resolvers being flooded with requests.



Tweet from Cloudflare's CTO on the flood of DNS traffic during the Facebook outage. (Twitter/@jgrahamc)

[Authoritative nameservers play a key role in DNS resolution](#), since they possess information on how to resolve a specific hostname under their authority.

### Having a quick response is key, as long as your badge is working!

How quickly you detect and get to the heart of an outage matters. Your runbooks also matter.

Sometimes fixing an escalation means you need to ensure your systems are different from one another. In this case, the badge systems your employees use to sign in and fix things should never be dependent on the thing you're trying to fix.

Troubleshooting in these types of instances is rarely straightforward. In Facebook's case, the symptoms were HTTP and DNS errors, which then impacted BGP.

### Update from Facebook's postmortem analysis

On October 5, 2021, the Facebook team released a very good post-mortem analysis of the incident.

The source of the incident was not caused by DNS or BGP, but by a maintenance routine job performed by Facebook staff aimed at assessing the availability of global backbone capacity. This backfired (unintentionally), taking down all the connections in their backbone network. Consequently, the Facebook routers couldn't speak to their data centers. This triggered a safety mechanism in which the BGP routes towards their DNS servers were withdrawn from the network, as we saw in our analysis.

Kudos to the Facebook team for the prompt recovery, but most importantly, for their transparency!

*Published on Oct 04, 2021*

## Incident Review: September 29-30, 2021 - Issues Caused by the Let's Encrypt DST Root CA X3 Expiration

By Sergey Katsev

As a monitoring and observability company, we have a lot of monitoring and observability built into our systems, as well. We have the standard monitoring programs in place to make sure that our systems are performing properly, data is flowing through our infrastructure, etc. At the same time, we continually observe for any sudden changes to tests that our customers are running.

On September 29, 2021, 19:22 UTC, we started to see a wave of alerts. The alerts originated from some of the web tests from our active observers, occurring when our Let's Encrypt "R3" certificate expired.

Another example of this happened in 2020 when the Sectigo AddTrust root certificate expired. Let's be clear, these types of incidents are pretty rare. The difference with this event was that a lot more servers rely on Let's Encrypt certificates.

The root cause of the crisis was not Catchpoint, our product, or any employee. Instead, it was an issue with changes to the certificate path by a certificate issuer. Furthermore, as we work with many vendors, we've received updates from some which indicate that solving this problem is as easy as downloading the latest OS updates. While this is true for some, it does not solve the problem in general!

Below we explain why, and how to solve it on the server-side so that all your clients can access your web service without issues. Let's dive into our incident review.

### Do you trust Let's Encrypt?

Before we get into the weeds, I just want to say that I, personally, trust Let's Encrypt. They're a great company that has made certificate management accessible to everyone, and they are extremely developer friendly. Additionally, partially because of them, the number of websites using encryption has skyrocketed in recent years.

Encryption is extremely important on the Internet. It's the basis for secure communications. Whether you're checking your bank balance, buying a new pair of socks from an e-tailer, or talking to your friends, you do so with the assumption that this transaction is secure.

In the last approximately eight years (2013-2020), the percentage of web pages using HTTPS has gone from 25% to more than 84%! There are several reasons for this incredible growth. One of them is that Google has been gradually forcing sites to use HTTPS by making HTTP-based sites "not secure." Still, no matter the cause, in that amount of time Let's Encrypt has gone from issuing certificates for about 50 million websites to over 230 million!



Increase in percentage of web pages loaded by Firefox using HTTPS. (Let's Encrypt)

At the same time, it doesn't matter that I trust Let's Encrypt if computers don't. That's exactly what happened on the evening of September 29 and again on the morning of September 30, 2021.

---

*"In the last year alone, Let's Encrypt have grown their market share quite a lot and as a CA becomes larger, its certificates enable more of the Web to operate and as a result, when something like this comes along, they have the potential to cause more problems. This is nothing to do with what Let's Encrypt have done, or have not done, this still comes down to the same underlying problem that devices out in the ecosystem aren't being updated as they should be."*

~Scott Helme, security researcher

---

## How digital certificate trust works

Here's a quick summary of how *certificate trust* works on the Internet:

- **Root certificates:** There are a handful of Root certificates. These are issued by major companies under a lot of scrutiny and are installed in the *Certificate Stores* of computers worldwide by the company that developed and maintains the OS. If you have a computer that can connect to an HTTPS website, you have such a certificate store. The MacOS laptop I'm writing this on has 161 "System Root certificates" installed.
- **Chain of Trust:** When someone launches a website nowadays, they must support HTTPS. Therefore, they purchase a certificate from a provider. There are many providers to choose from. Some have their own Root certificate and others have a *Certificate Authority* certificate, which was signed by one of the Root certificates or by another *Certificate Authority*. In this way, there is a *chain of trust* from the website's certificate all the way to the root certificate.
- **Intermediate certificates:** When you go to the HTTPS website, the server hosting the website sends the certificate during the SSL handshake with the client (browser/HTTP client). It might also send you one or more *intermediate certificates*. These intermediate certificates are what it thinks you might need to connect the chain of trust from the server certificate to one of the root certificates you installed on your computer.
- **Validation process:** Your browser "walks" the chain of trust, from the server certificate up to the root. If it makes it all the way to the root and finds it in its "store," the chain is validated, and the connection is allowed to proceed. Otherwise, you get a security warning like the one below.

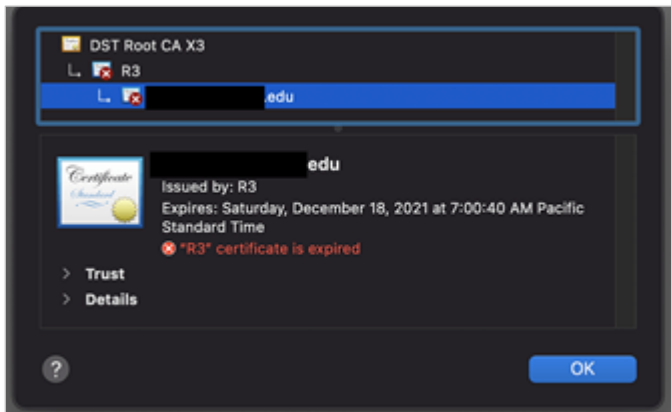


## R3 certificate expiry and the chain of trust

Let's go through the details of what happened.

As mentioned, the first problems we saw with web tests from our synthetic nodes began at 19:22 UTC on September 29, when the Let's Encrypt "R3" certificate expired. Here's the certificate information for this intermediate certificate: <https://crt.sh/?id=3479778542>.

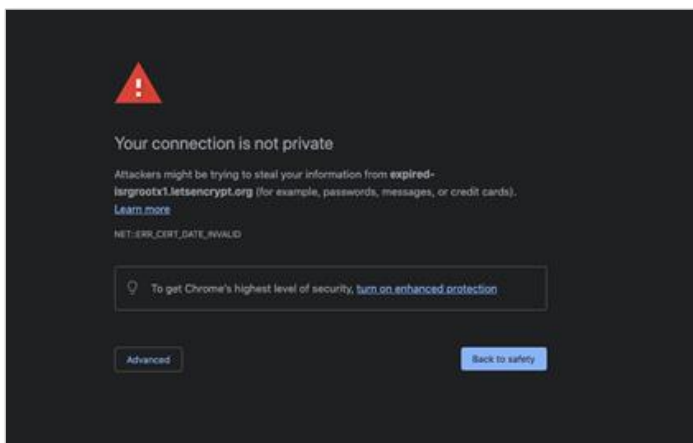
The most important piece of information here is the expiration date. Since this is an intermediate certificate, this means that Let's Encrypt used this certificate to sign other certificates for their customers. For example, see the following screen capture.



DST Root CA X3 expiration date on an intermediate certificate. (Let's Encrypt)

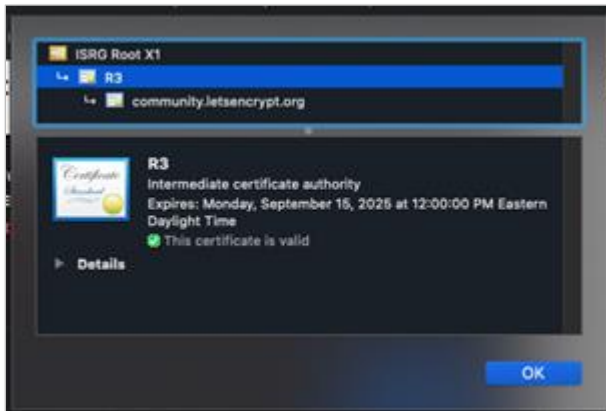
This screenshot shows a web site whose certificate was signed with this R3 certificate. As soon as the certificate expired, this website was no longer accessible!

A browser which tried to validate the website's certificate would walk the chain of trust and find the intermediate certificate expired. That's when you get scary looking errors in your browser, like the one below.



Browser message: "Your connection is not private." (Let's Encrypt)

Let's Encrypt published a new R3 certificate! The new expiration date is in 2025 – plenty far away. Everything's great, right? Well, not quite.



New Let's Encrypt R3 certificate. (Let's Encrypt)

It turns out that the certificate needs to be updated in your computer – usually through a Windows or MacOS update – before it works. A lot of people don't update their computers as often as they should, though. Even worse, a lot of embedded devices rarely if ever update their certificates! Someone here at Catchpoint mentioned that his kids couldn't watch their favorite streaming show from his SmartTV because of this issue!

Let's say you got the latest R3 intermediate certificate installed in your whole fleet of devices. Now you can go to any of those millions of sites with Let's Encrypt certificates, right? Well, sort of...until 14:00 UTC the following day.

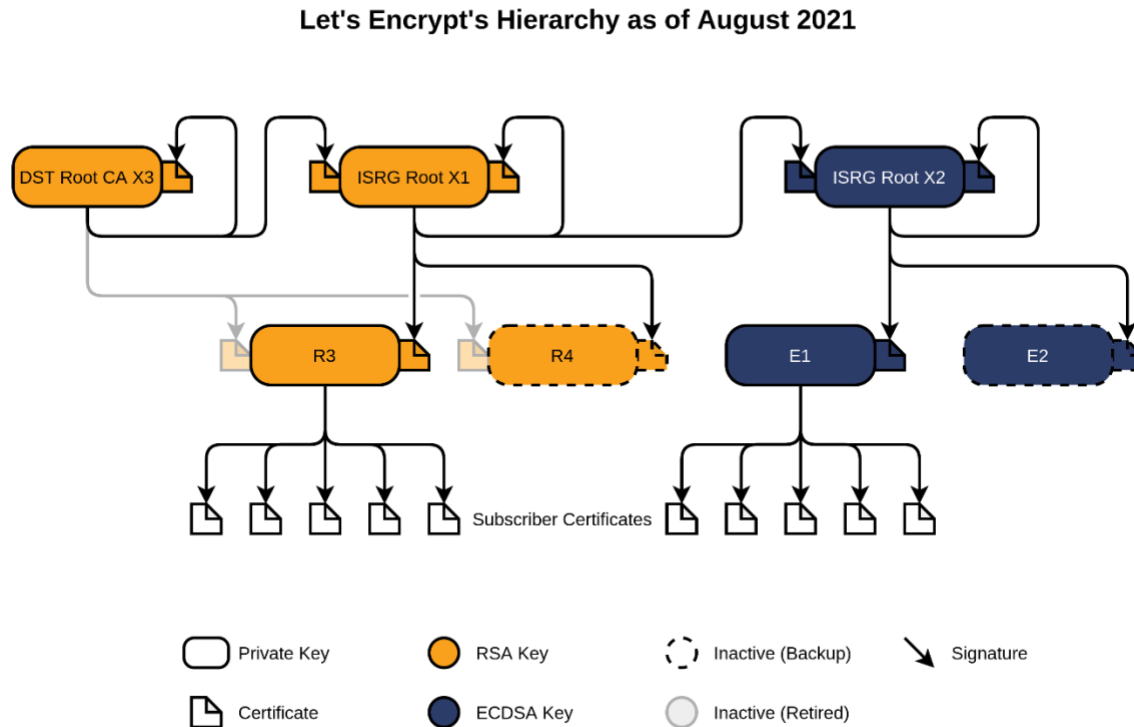
### **September 30, 14:00 UTC: DST root CAx3 certificate expiry and its consequences**

At 14:00 UTC on September 30, the DST Root CA X3 certificate expired. The details are a little confusing. Bear with me.

Originally, the DST Root CA X3 was used to sign all Let's Encrypt certificates (including the R3 intermediate certificate above). Let's Encrypt also cross-signed the certificates using their own ISRG Root X1 certificate. This was done because the DST certificate was already present in most browsers and devices. However, the ISRG certificate was not.

As Let's Encrypt became more well-known and the ISRG certificate was available in all major devices, they stopped relying on the DST certificate.

The following diagram portrays the certificate hierarchy directly from Let's Encrypt.



Let's Encrypt Certificate Hierarchy (Let's Encrypt)

Note that any server certificates ("Subscriber Certificates") that were signed by R3 were signed by either DST root or ISRG root, or, most likely, cross-signed by both.

Here's the DST root certificate's information: <https://crt.sh/?id=8395>.

There are actually two versions of the ISRT Root X1 certificate: <https://crt.sh/?id=3958242236> and <https://crt.sh/?id=9314791>.

The second one is self-signed. This is fine for a Root CA certificate which is present in most devices around the world. The first one, though, is signed by DST Root CA X3!

When the DST root certificate expired, this caused problems for two classes of system:

Systems which didn't have an updated copy of the ISRT Root X1 certificate started failing to connect to sites using Let's Encrypt because their site certificate was signed by R3, which was signed by ISRG, which was signed by DST - which had expired!

Systems which did have the proper updated copy of the ISRT Root X1 certificate but wanted to validate the DST Root certificate anyway, because it had cross signed the R3 certificate!

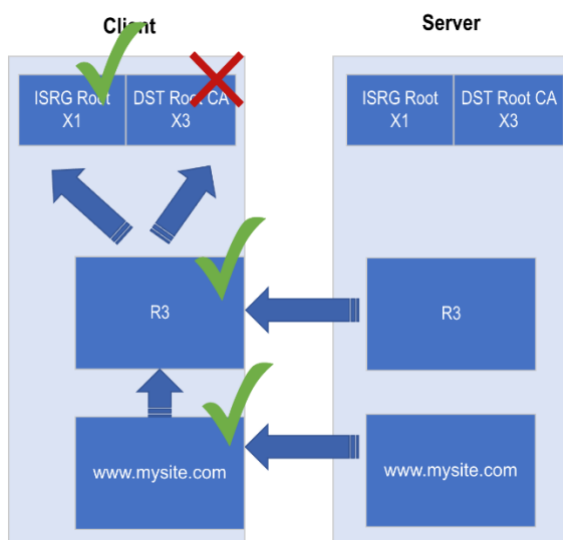
## Fixes, fixes...

The first category was relatively easy to fix: update the OS or download the new certificate and install it, assuming it's not an embedded device that hasn't issued an update.

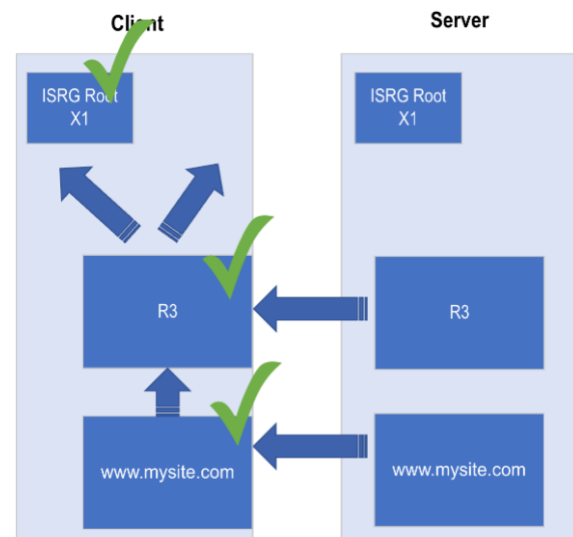
The second one is harder. For example, any software which relies on OpenSSL 1.0.2 or earlier will have this problem - and there's no way for the client to fix it.

Think of it this way:

Usually, your website sends the server certificate and any intermediate certificates that the client might need. As the client walks the chain of certificates, it sees the site certificate signed by R3. Then it sees that R3 is signed by ISRG, but also by DST - some browsers or other HTTP clients only validate one. They then find that the ISRG certificate is valid, and they're satisfied. Others need both to be valid but they're not, because the DST certificate is expired!



Client certificate validation process when the server includes DST certificate (Catchpoint)



Client certificate validation process when the server doesn't include DST certificate. (Catchpoint)

If you run a website and want customers to be able to connect from devices such as these, there's only one fix: Regenerate the certificate that your site uses so that it is no longer cross-signed by the DST certificate. At that point, your server will stop sending it to the client to validate, and the client won't fail to validate it.

This is particularly important if you have HTTPS-based services that aren't being accessed directly by browsers on a laptop. Maybe you're serving RSS feeds or have an API accessed by embedded devices. Maybe your clients access the site through a proxy (there's a higher chance that some of the users trying to access your server are unable to due to this problem).

## Fixing this on your server

The first step to fixing this problem was understanding the impact on your customers.

The second step, which many system administrators don't think about, was understanding who your customers are! This is a particularly "weird" issue because there's no way to resolve it for *everyone*, except I guess switching from Let's Encrypt.

With Let's Encrypt, no matter what you do, someone will be impacted. You have to choose whether your clients are likely to be using old versions of Android (old phones, but also devices like smart TVs), or using old versions of OpenSSL (many other embedded network devices). Or maybe you have a lot of users with non-updated Operating Systems.

Because of the way the certificate chain was put together by Let's Encrypt, they put the onus on each server administrator to decide who to support and who to break.

## In conclusion...

Here at Catchpoint, we have a huge footprint of test agents in every corner of the world. These agents have different configurations of hardware, software, firmware, etc. Because of this large fleet, we saw and solved almost every flavor of the issues described above.

However, the customers accessing your website probably don't have a 24/7 Operations team. In theory, we can update all our agents that act as clients to connect to Let's Encrypt-based sites properly or otherwise ignore this server misconfiguration issue. At the same time, the reality is that you cannot expect every user in the world to do this - and often they cannot.

As the owner of that service, you have the solution within your control on the server side, so it is on you to fix it.

*Published on Oct 01, 2021*

## Incident Review: October 1, 2021 – An Account of DNS Misconfiguration at Slack

By Karthik Suresh

DNS observability is an essential part of any Ops team's strategy. Looking for proof? You don't need to go any further than the October 1, 2021, Slack outage.

This was a busy week for Ops teams across the globe. Many were forced to urgently rotate SSL certificates after one of Lets Encrypt's root certificates expired. Collaboration plays a critical role during such situations where members in a team or multiple teams must communicate and work with each other to complete a collective task rapidly and efficiently.

Unfortunately, things got more challenging as one of the world's largest collaboration and messaging applications, Slack, was not accessible for various users worldwide during the same period.

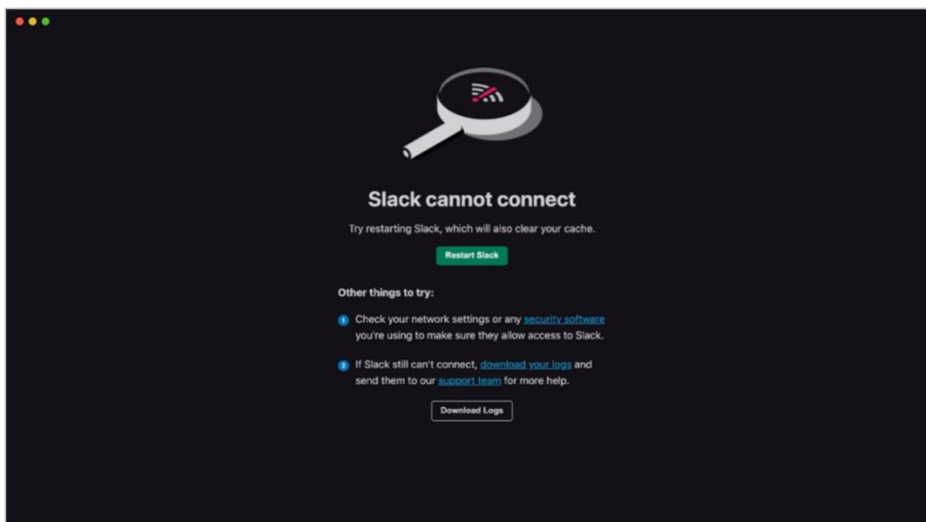
DNS misconfiguration was at the core of the issue at Slack. If the process of DNS resolution fails, users experience outages like this. However, you can take action to avoid business impact.

Let's start by breaking down the issue.

### Slack acknowledges the issue

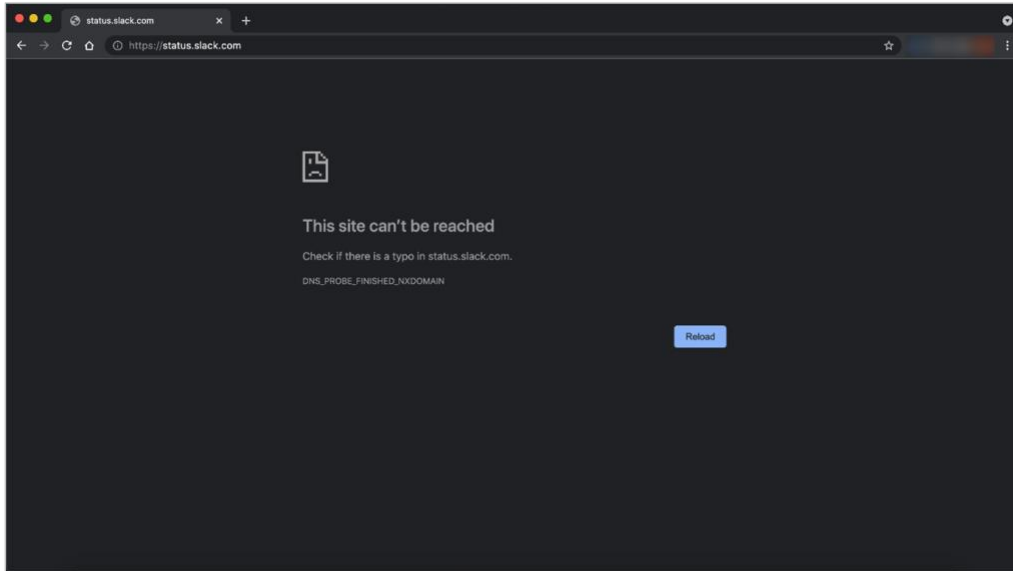
Users were not able to access desktop, mobile, and web applications of Slack from 15:30 AM UTC onwards. This was due to a DNS failure, which was later [acknowledged by Slack](#).

At the time of the publication of this article, the issue was still ongoing for some users and at 06:57 UTC, Slack [announced it may take up to 24 hours](#) to completely resolve this issue for all users. The issue was resolved at around 22:00 UTC.



Slack error message (Slack)

While the outage was underway, users were struggling to understand if they were not able to access Slack due to their device, wireless network, or ISP connectivity. Things got more difficult as Slack's status page was down due to the same issue.



Slack status page is down (Slack)

## Why monitoring from the cloud isn't enough

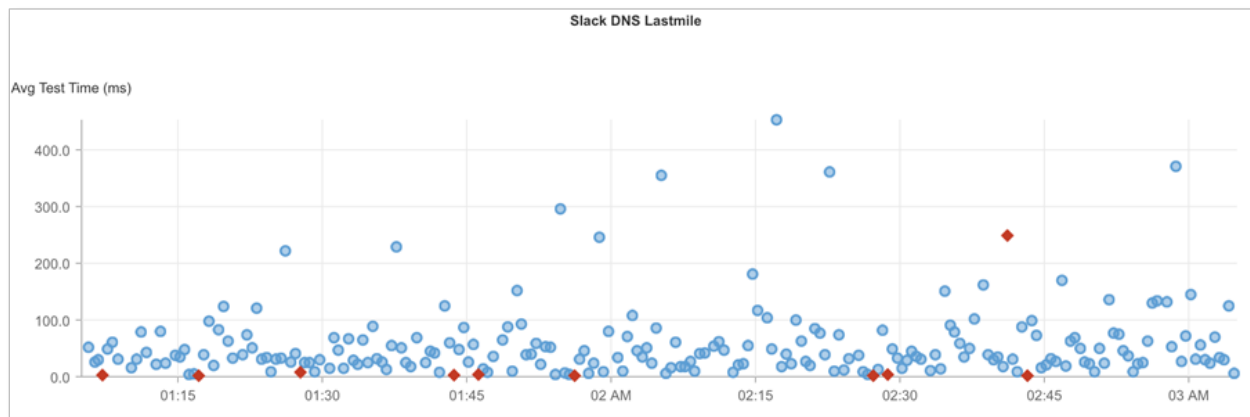
During such incidents where Operation teams are not able to collaborate efficiently with each other, things can easily spin out of control and cause outages that directly impact customers. While IT teams in some organizations might already monitor their SaaS applications, it is not surprising if only a handful had triggered any alarms for Slack.

Adding to the challenge is the fact that most monitoring and observability solutions are hosted on cloud instances. [Monitoring applications from cloud instances, however, leaves dangerous blind spots and does not accurately represent end user experience.](#)

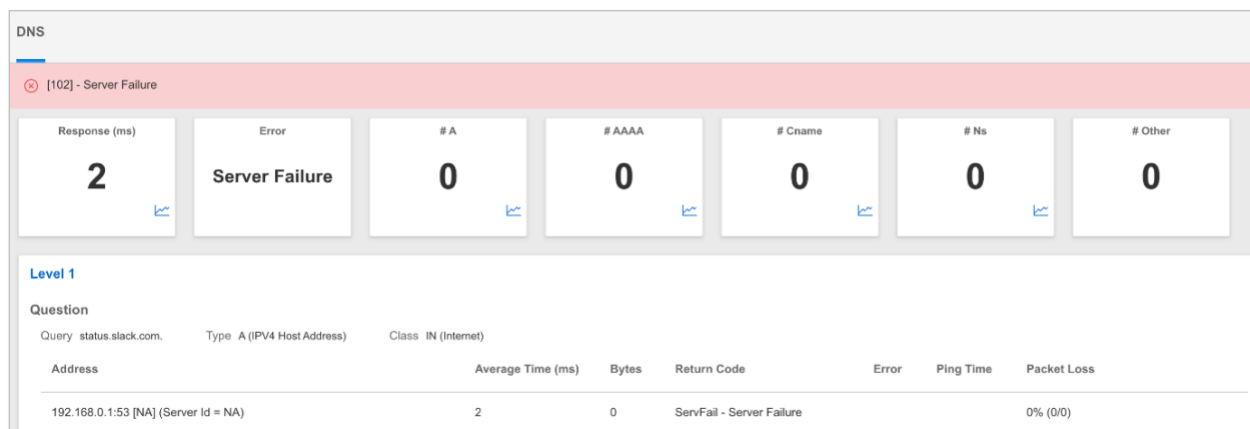
A good monitoring and observability strategy must include a combination of observation across backbone and last mile networks. The backbone network has predefined bandwidth and consistent network connectivity. This allows you to monitor, measure, and benchmark application performance without any network fluctuations. At the same time, the last mile network represents availability and performance for your real end users trying to access digital services on their home/office networks.

## Catchpoint's last mile tests detected DNS issues as the root cause

Catchpoint's last mile tests detected Slack DNS issues, allowing our platform to proactively notify the impacted IT teams.



Scatterplot data showing intermittent failures for Slack DNS tests (Catchpoint)



DNS records showing server failure while resolving slack.com domain (Catchpoint)

Even after 15 hours of the outage, some users were unable to access Slack. Those who were aware of the issue and its root cause, however, fared quite differently. Armed with an understanding of what was going on, they were able to quickly mitigate the problem by overriding their default DNS resolver with a public DNS resolver, such as 8.8.8.8 or 1.1.1.1.

Slack has [since confirmed the outage](#) was, "caused by our own change and not related to any third-party DNS software and services." It was related to Slack's TTL allowing for caching of responses for up to two days.

The lesson here? DNS might be a small service in the delivery chain, but minor mistakes in configuration can take hours to recover if you have large TTL for your records. Read more about [how TTL can impact DNS responses](#).



## Understand how to resolve DNS issues more quickly

DNS is at the core of the Internet. If the process of DNS resolution fails, users will experience outages such as this one. Observing the DNS of all your essential SaaS services from the cloud, backbone and last mile is essential to understanding the true performance of DNS.

The fix itself is easy, but only if you know what needs to be fixed!

*Published on Oct 01, 2021*

---

*"What's so bad about cascading failures? They take down your whole service, most of the time, because the effects of them spread, unless you have some sharding approach, perhaps. They don't self-heal. Once you're in this cycle, it stays that way until you intervene. You don't really get warning of them. You can think you're fine, everything looks healthy, then you're on that cliff edge, and you just step over it."*

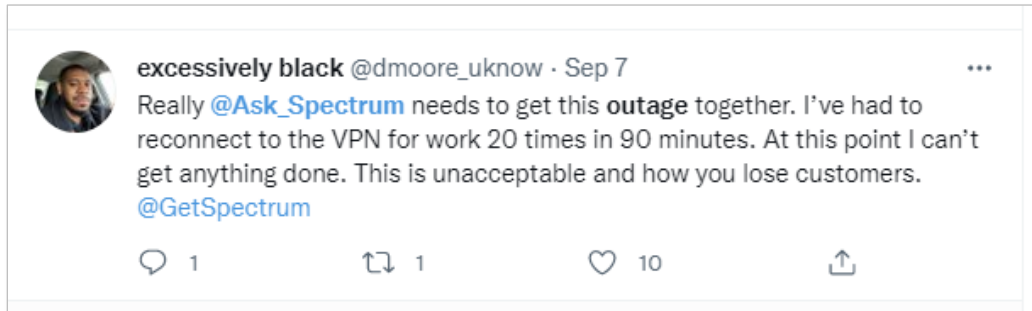
*~Laura Nolan, SRE, Slack*

---

## Incident Review – September 7, 2021: A Case of Dr. BGP Hijack or Mr. BGP Mistake at Spectrum?

By Alessandro Improta and Luca Sani

September 7, 2021, 16:36 UTC: [an outage hit Spectrum cable customers in the Midwest](#) of the U.S., including Ohio, Wisconsin, and Kentucky. Users of their broadband and TV services hit social media to voice their annoyance at the disruption it was causing.



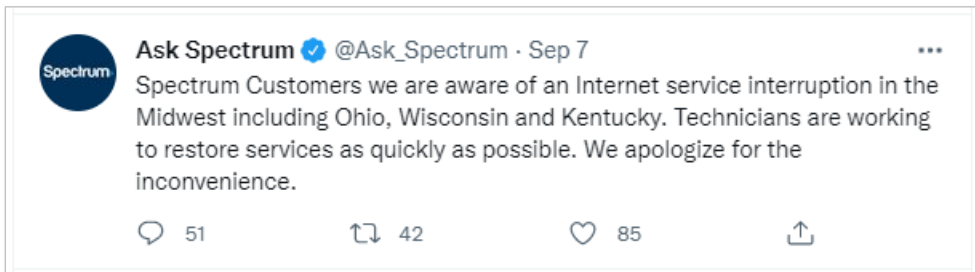
Spectrum user complaint on Twitter (Twitter/@dmoore\_uknow)



Spectrum user complaint on Twitter (Twitter/@BrettLarter)

Everything was resolved at around 18:11 UTC, and services were restored to users.

## Fact checking the Spectrum investigation



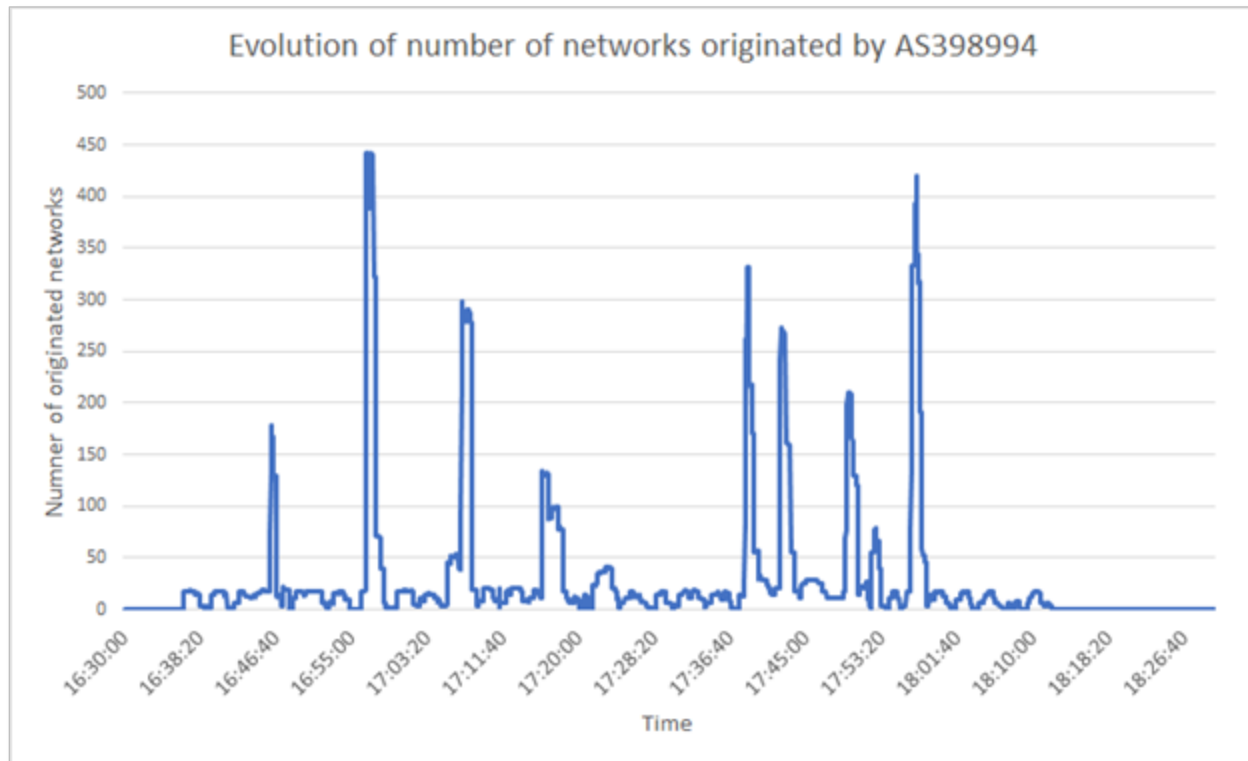
Official Spectrum Tweet acknowledging outage (Twitter/@Ask\_Spectrum)

During the outage, Spectrum was vague about [the type of issue they were having](#) that was causing it. Our investigation, however, showed it may have involved a BGP route hijack.

A [BGP route hijack](#) happens when an autonomous system (AS) claims to be the origin for a network that has been assigned to another AS. If the hijack is accidental, it can lead to a denial of services. If deliberate, at its worst, another AS could attempt to steal sensitive information.

To better understand what happened, we investigated data collected by rrc10, the RIS route collector deployed by RIPE Network Coordination Centre (NCC) at the Milan Internet Exchange (MIX), in Italy. Looking at the closest RIB snapshot available, we could see that Spectrum (AS10796) was announcing 690 networks, most of which routed via its backbone (AS7843).

From the update files provided by rrc10 collector between 16:30 and 18:30 UTC, we could further see that KHS USA (AS398994) started to originate 449 of the 690 networks previously originated by Spectrum. This was what started the outage. Routes were seen by most of the RIS peers up to 18:11 UTC, when KHS stopped announcing any routes.



Update files provided by rrc10 collector between 16:30 and 18:30 UTC (Milan Internet Exchange)

### So... was it Dr. BGP hijack or Mr. BGP mistake?

One of the peculiarities of this hijack is that it started in the belly of the attacked AS. KHS announced the Spectrum networks via Spectrum AS itself (AS10796), and the Spectrum backbone (AS7843) propagated them in the wild. Only a few routes were announced via Spectrum AS itself (AS10796) and Tata communications (AS6453), one of Spectrum's providers.

```
[luca@localhost rrc10]$ bgpscanmer -p "398994" updates.20210907.1635.gz | grep -v "#"
```

```
+66.198.14.0/24|24482.6453.10796.398994|217.29.66.158|1||6453:1000.6453:1200.6453:1202.6453:10796.24482:1.24482:12020.24482:12021.24482:20300.24482:64601|217.29.66.158.24482:1631032582|1
```

```
+24.100.44.0/23.209.97.76.0/22.76.11.148.0/22.209.97.72.0/23.72.51.192.0/19.24.100.48.0/22.76.11.248.0/22.24.100.32.0/22.72.51.192.0/19.72.51.250.0/23.76.11.128.0/22.72.51.236.0/22.209.97.68.0/22.76.11.226.0/24.72.51.140.0/22.168.250.32.0/20|12637.3257.7843.10796.398994|217.29.66.66|e||3257:4000.3257:8156.3257:50002.3257:50120.3257:51100.3257:51102.12637:65020.12637:65021|217.29.66.66.12637|1631032590|1
```

```
+24.100.48.0/22.72.51.192.0/19.168.250.32.0/20.24.100.176.0/21.209.97.72.0/23.72.51.132.0/23.76.11.248.0/22.24.100.44.0/23.72.51.140.0/22.209.97.68.0/22.72.51.250.0/23.24.100.32.0/22.209.97.76.0/22.76.11.22.0/24.72.51.192.0/19.76.11.148.0/22.76.11.128.0/22.72.51.236.0/22|201333.6762.3257.7843.10796.398994|217.29.67.74|7||6762:30.6762:13000.201333:100:6762|217.29.67.74.201333|1631032592|1
```

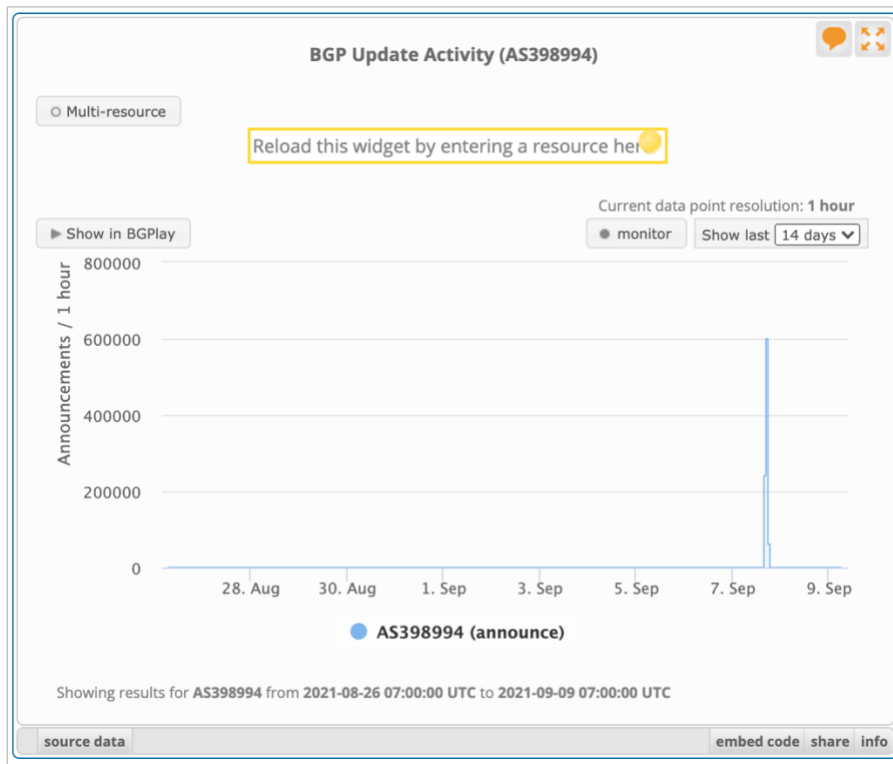
```
+24.100.40.0/22.72.51.160.0/19.168.250.32.0/20.24.100.176.0/21.209.97.72.0/23.72.51.132.0/23.76.11.248.0/22.24.100.44.0/23.72.51.140.0/22.209.97.68.0/22.72.51.250.0/23.24.100.32.0/22.209.97.76.0/22.76.11.22.0/24.72.51.192.0/19.76.11.148.0/22.76.11.128.0/22.72.51.236.0/22|201333.3257.7843.10796.398994|217.29.67.74|7||3257:4000.3257:8156.3257:50002.3257:50120.3257:51100.3257:51102.201333:100:3257|217.29.67.74.201333|1631032592|1
```

```
+72.51.236.0/22.24.100.176.0/21.72.51.140.0/22.76.11.148.0/22.76.11.226.0/24.76.11.248.0/22.24.100.32.0/22.72.51.192.0/19.72.51.132.0/23.24.100.44.0/23.209.97.68.0/22.209.97.72.0/23.76.11.128.0/22.24.100.48.0/22.209.97.76.0/22.72.51.250.0/23.72.51.160.0/19.168.250.32.0/20|12779.3257.7843.10796.398994|217.29.66.65|7||12779:10111.12779:65096|217.29.66.65.12779|1631032598|1
```

Examples of BGP events carrying hijacked subnets, as collected by rrc10 collector. (Milan Internet Exchange)

Another oddity is that KHS didn't announce any networks before or after the hijack (this can be seen from the BGP Update activity widget provided by [RIPE Stat](#)).

What caused the issue? The outage may have been caused by an experiment at Spectrum. Alternatively, similar scenarios can happen if someone exploits a security vulnerability inside the carrier and finds a way to open a BGP session with one of their routers. This could then lead to a hijacking of the routes to create an outage on purpose. In the past, we have seen similar scenarios where this has been the case, such as those [listed here](#).



BGP Update activity widget. (RIPE Stat)

*"BGP hijacking may be the result of a configuration mistake or a malicious act; in either case it is an attack on the common routing system that we all use... The problem is, BGP was created long before security was a major concern. BGP assumes that all networks are trustworthy. Technically, there are no built-in security mechanisms to validate that routes are legitimate."*

~Megan Kruse, Director, Partner Engagement and Communications, Internet Society

## Lessons for network administrators

Either way, there are a few lessons network administrators can obtain from what happened.

First, around BGP: While a BGP hijack may not have been the case with Spectrum, it is worth mentioning that unrecognized sources absolutely must not be able to set up a BGP session on their own and announce networks at will.

Second, network administrators need to set up automated controls to drop any route announcement related to networks that their customers are not allowed to announce. We suspect that this level of control was missing at the Spectrum router. Otherwise, AS 398994 would not have been able to hijack routes belonging to 449 Spectrum networks.

However, simple controls are not enough for transit AS. If any of them were setting up a list of networks that AS10796 was allowed to announce, that would still not have stopped the spreading of the hijack in the wild. Indeed, AS10796 was the original owner of the networks and signed most of its routes in RPKI, including 431 networks out of the 449 being hijacked.

Dropping routes found invalid via RPKI checks would have been a solution to mitigate the hijack spread. However, even that would not have stopped the hijack itself.

Third, but not least, network administrators must take measures against events like this one by investing in 24/7 BGP monitoring tools. At Catchpoint, we inform customers about hijacks within only a few seconds.

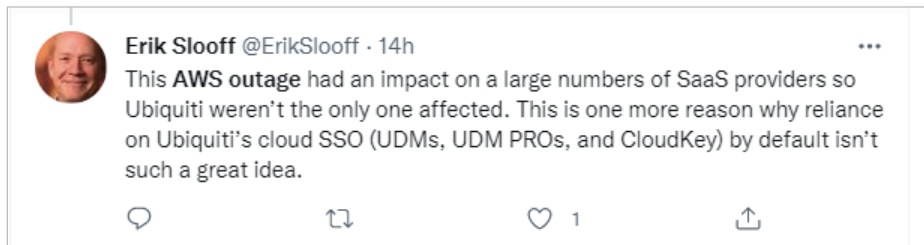
*Published on Sep 09, 2021*

## Incident Review: August 31, 2021 – AWS Services Hit by Major Spikes in Response Times

By Navya Dwarakanath and Anna Jones (with contributions from Siva Dwivedula)

On Tuesday, August 31, 2021, AWS users across large parts of the West coast (US-West-2 region) were impacted by major spikes in response time. Some of AWS' most critical services were affected, including Lambda and Kinesis. The incident lasted for approximately four hours, creating widespread headaches – from websites being down to difficulties logging into applications – across the entire US-West-2 region.

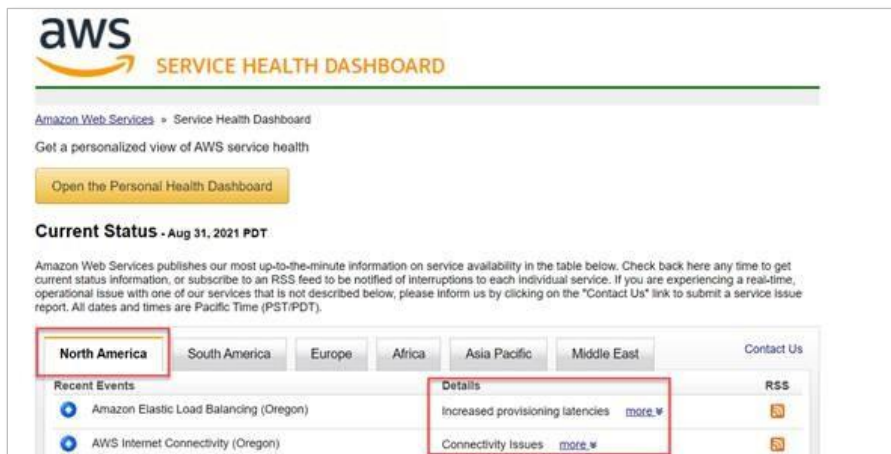
Companies, developers, and DevOps teams shared their angst on social media and news sites. Those commenting on the incident included [The Seattle Times](#), major gaming company [Zwift](#), and SaaS platform [Ubiquiti](#).



Twitter chatter on the impact of the AWS outage. (Twitter/@ErikSlooff)

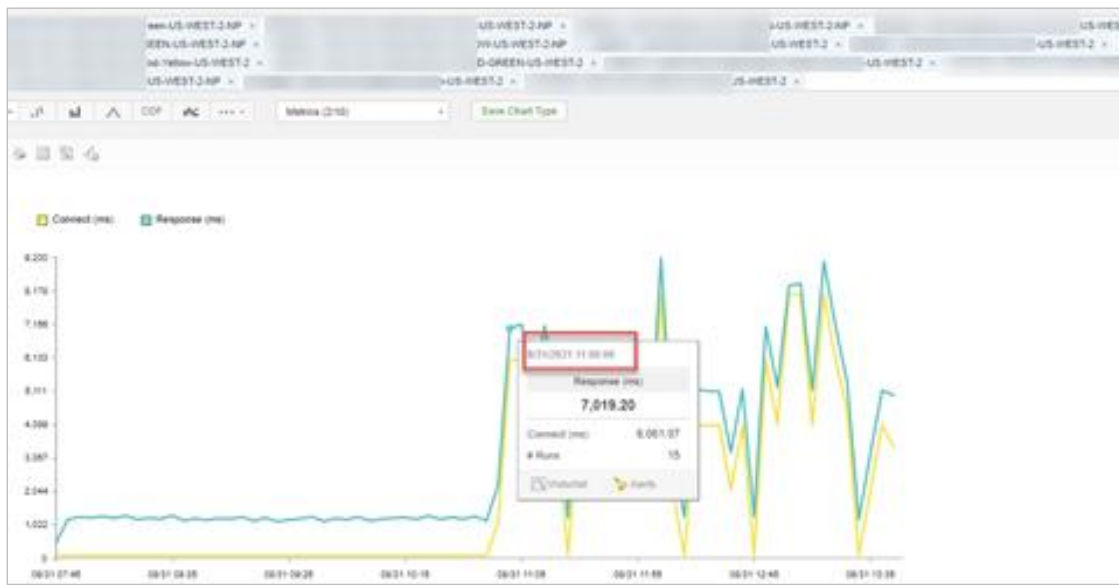
### What the AWS status dashboard showed

The AWS Service Health Dashboard revealed increased provisioning latencies to Amazon Elastic Load Balancing in Oregon and AWS Internet connectivity issues in the same region.



AWS Status Report. (AWS)

Impacted AWS services included Lambda, ELB, Kinesis, RDS, CloudWatch, and ECS.



Catchpoint data revealed a major spike in response times for applications using AWS Services. (Catchpoint)

Only users in the US-WEST-2 were impacted, meaning Oregon.

### Root cause identified: network connectivity issues

At 9:26PM UTC, AWS identified the root cause of the issue as, “a component within the subsystem responsible for the processing of network packets for Network Load Balancer.” This led to impairment of the NT Gateway and PrivateLink services, “no longer processing health checks successfully,” followed by further performance degradation.

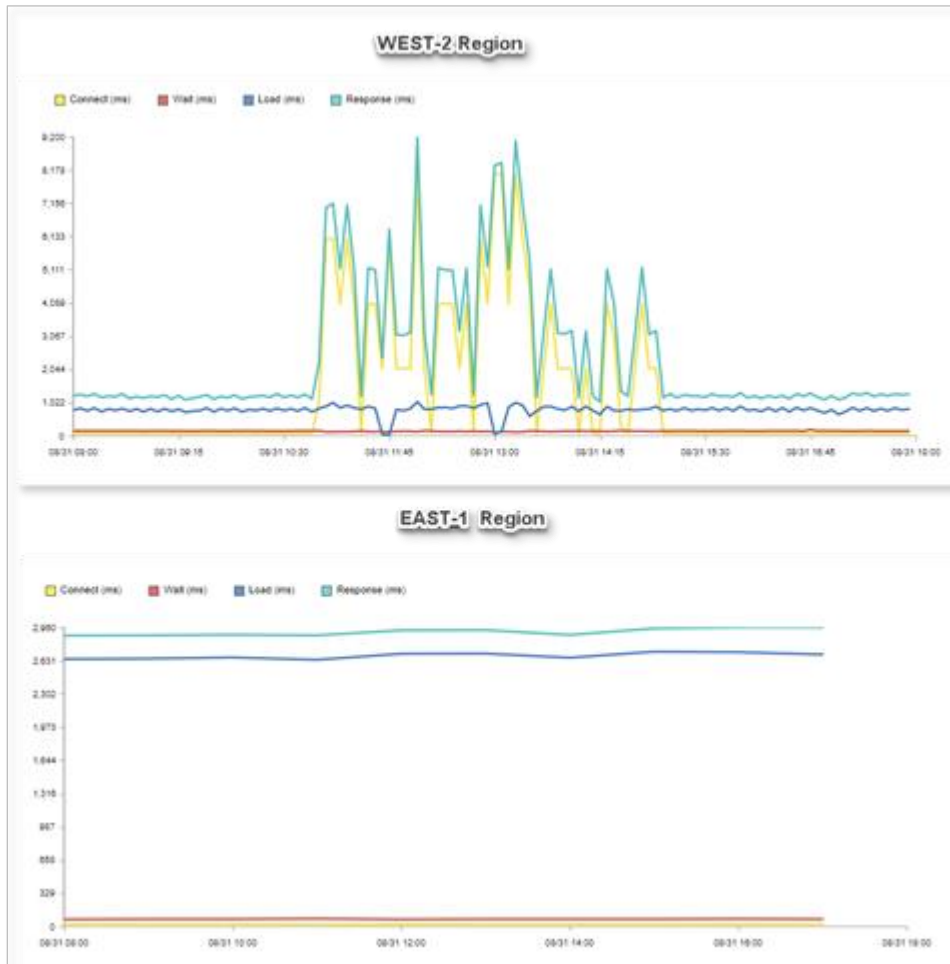
2:26 PM PDT We have identified the root cause of the issue affecting network connectivity within a single Availability Zone (usw2-az2) in the US-WEST-2 Region and are actively working on mitigation. A component within the subsystem responsible for the processing of network packets for Network Load Balancer, NAT Gateway and PrivateLink services became impaired and was no longer processing health checks successfully. This resulted in other components no longer accepting new connection requests, as well as elevated packet loss for Network Load Balancer, NAT Gateway and PrivateLink endpoints. For immediate mitigation for NLB, customers are able to: 1) disable the subnet for usw2-az2 in Network Load Balancer 2) create a new Network Load Balancer that does not use usw2-az2. For NAT Gateway/PrivateLink, you may modify your route tables to direct traffic to NAT Gateways in other Availability Zones or disabling PrivateLink endpoints in usw2-az2.

AWS root cause statement. (AWS)



Using Catchpoint's dataset, we were able to draw on additional metrics to validate that the cause of the outage was indeed a network connectivity issue. Our 50+ metrics allow you to narrow down issues to a specific component, so that you can then answer the question, "Is it the network or the application that is causing the problem?"

In this case, you can see below that the overall response time spiked because of an increase in connect time to the servers, which is impacted by the network. However, the load and wait times, which are related to the server processing time – and hence indicative of applications/server-side issues – was flat with no spikes.







Comparison of AWS response times between WEST-2 and EAST-1 regions. (Catchpoint)

Comparing this to the US-EAST region, we can clearly see that the issue was concentrated in US-WEST.

## Catchpoint detects and alerts on AWS outages first

Catchpoint first detected issues for our customers at 18:00 UTC on Tuesday August 31<sup>st</sup>. Our data analysis revealed widespread connectivity failures in the U.S.-West region.

We immediately triggered our first alert – a full 25 minutes before AWS acknowledged the issue (AWS' first mention on their status page that they were investigating the issue took place at 18:25 UTC).

Recent Events	Details	RSS
 Amazon Elastic Load Balancing (Oregon)	Increased provisioning latencies <a href="#">more</a> ▾	
 AWS Internet Connectivity (Oregon)	Connectivity Issues <a href="#">less</a> ▲	
<div> <div>11:25 AM PDT</div> <div>We are investigating an issue which is affecting network traffic for some customers using AWS services in the <b>US-WEST-2 Region</b>.</div> </div> <div> <div>12:13 PM PDT</div> <div>We continue to investigate the issue affecting network connectivity within a single Availability Zone (usw2-az2) in the US-WEST-2 Region. While we continue to work towards root cause, we believe that the issue is affecting connectivity to Network Load Balancers from EC2 instances, connectivity from Lambda to EC2 instances and other AWS services, as well as connectivity between EC2 and some AWS services using PrivateLink. In an effort to further mitigate the impact, we are shifting some services and network flows away from the affected Availability Zone to mitigate the impact.</div> </div> <div> <div>1:00 PM PDT</div> <div>We continue to investigate the issue affecting network connectivity within a single Availability Zone (usw2-az2) in the <b>US-WEST-2 Region</b>. We have narrowed down the issue to an increase in <b>packet loss</b> within the subsystem responsible for the processing of network packets for Network Load Balancer, NAT Gateway and PrivateLink services. The issue continues to only affect the single Availability Zone (usw2-az2) within the US-WEST-2 region, so shifting traffic away from Networking Load Balancer and NAT Gateway within the affected Availability Zone can mitigate the impact. Some other AWS services, including Lambda, ELB, Kinesis, SQS, RDS, CloudWatch and ECS, are seeing impact as a result of this issue.</div> </div>		

AWS Status update. (AWS)

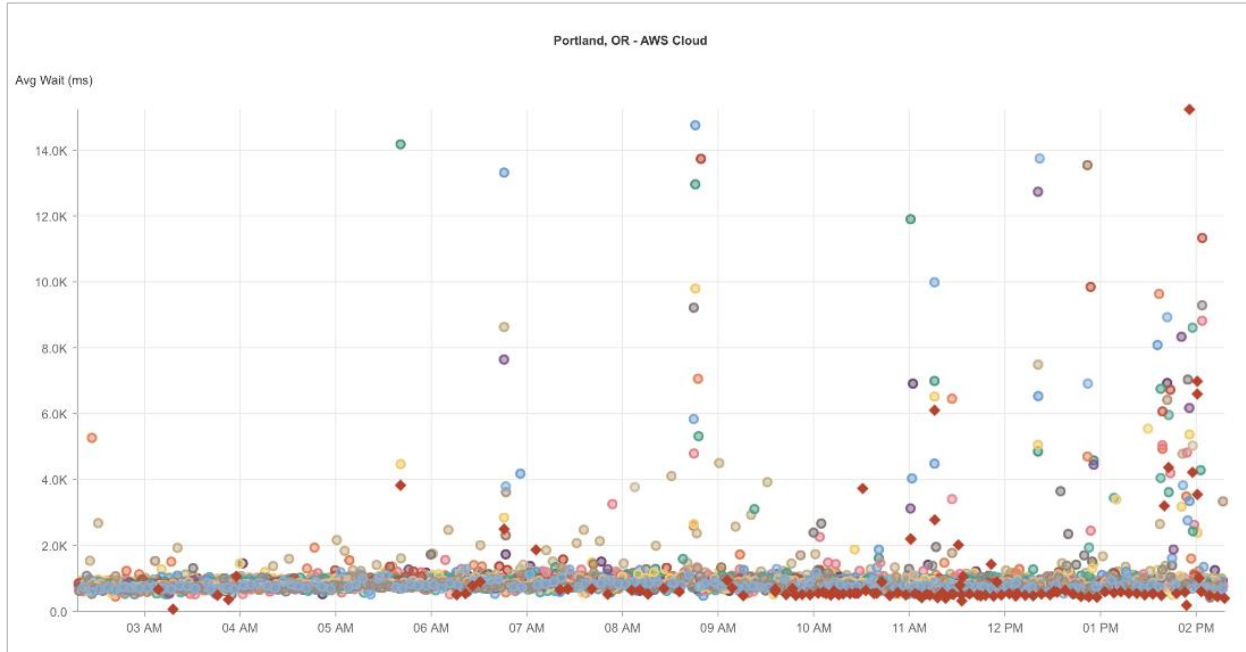
It was a similar story with DDoS issues at AWS two years ago [when we detected the issue five hours ahead of them](#). At the time, when one of our top customers reached out to AWS support about the problem straight after being alerted by Catchpoint, they were unaware there was an incident going on.

Why was this the case? Unlike other observability platforms, Catchpoint is not hosted on a cloud provider, so when a cloud provider has an incident impacting their solutions, we are not impacted. Our platform will continue to work, alerting you as soon as we detect any problem.

## The impact of AWS' outage on active observability services running on AWS

Just to be clear, this outage not only impacted AWS, but any monitoring or observability services running on AWS. This outage serves as a reminder for organizations to evaluate and verify their own infrastructure setup, including monitoring, observability, and failover strategies.

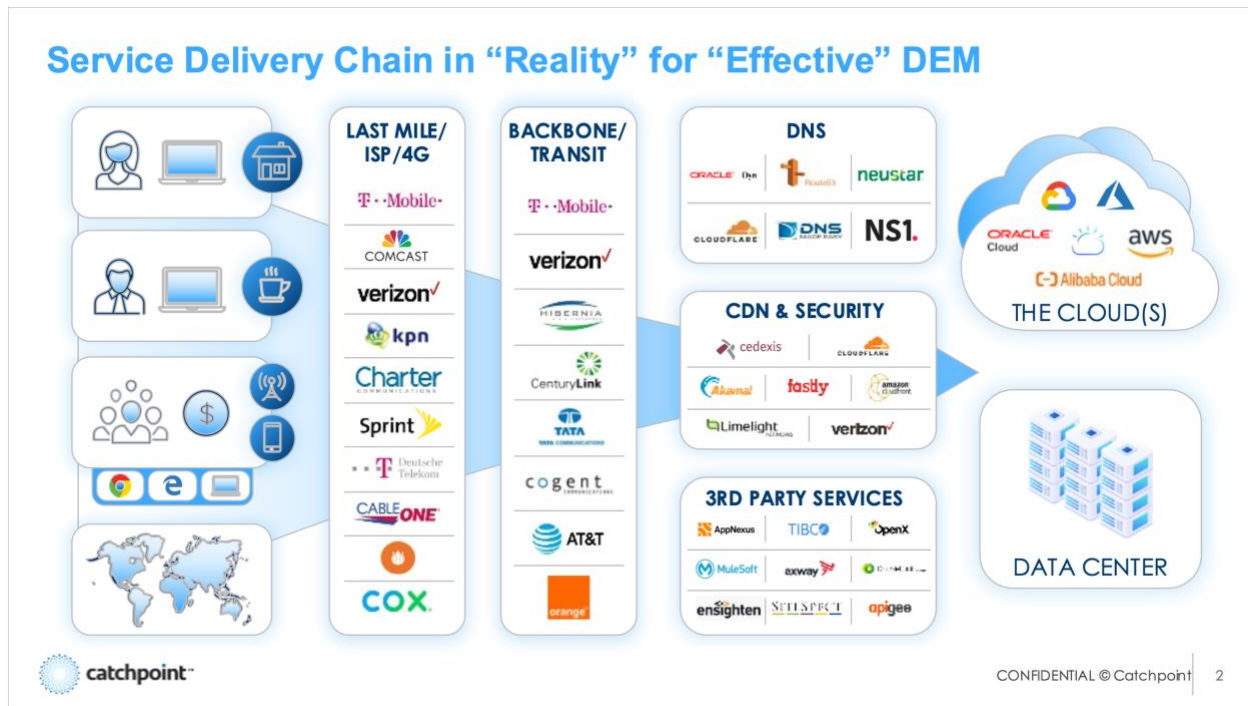
It's also worth taking a moment to ensure that you don't rely on cloud-only monitoring strategies, which as we've seen, can lead to blind spots.



Catchpoint data showing that cloud-only monitoring causes false negatives. (Catchpoint)

The image above shows the data from the Portland AWS observer (the affected region) and observed spikes in response times. When you are observing from locations only on the cloud and the cloud provider has an issue, your tool will make it seem look like you have a problem.

In other words, if you were observing from this region but didn't have any services or infrastructure hosted there, you might still have received alerts telling you that you had a problem. These are false negatives, meaning your on-call teams are getting pinged with unnecessary alerts that can waste valuable time and resources.



Service delivery chain for effective DEM. (Catchpoint)

You can reduce the noise by deploying a holistic monitoring and observability strategy. Catchpoint has the industry's largest network of cloud observers, but we also simulate the entire end user experience. In other words, we have observers on local ISPs, major backbones, mobile networks, and the clouds that your end users connect to when visiting a site or using an app. This allows us to detect outages and performance issues from anywhere, in real time.

### How can you prevent single points of failure?

Unlike Google, the founders of SRE, most companies rely on other providers like AWS and GCP, along with a CDN (or CDNs), for their infrastructure and services. This means that not only do you need Service Level Indicators (SLIs) and Service Level Objectives (SLOs) for your applications and services, but you also need to take a close look at your providers and vendors, particularly anything in your infrastructure that is a single point of failure. This is because SLOs and SLIs for most companies will have a dependency on the vendors and providers being used. If you are hosted in the cloud, the cloud vendor having an issue could be analogous to you missing your SLOs.

By monitoring your vendor SLOs (from beyond the cloud), you can understand their impact on your SLOs and system architecture to properly deliver the level of experience you are aiming for.

*Published on Sep 03, 2021*

## Incident Review: August 31, 2021 – High CDN Network Response Times Slow Down Major Websites Worldwide

By Ahamed Ali

In August and July 2021, we saw several outages and performance degradations from some of the world's most widely used CDNs. For example, there was the June 8, 2021, Fastly outage (owing to DNS or configuration issues) and an Akamai outage on July 22, 2021 (also likely caused by DNS failure).

Let's take a look at why these incidents occurred.

### Performance issues create problems at Akamai

On Tuesday, August 31, 2021, we saw problems at Akamai due to a performance issue. It wasn't an outage that showed failures or error pages for end users, but a problem which slowed down the performance of the websites that Akamai supports.

You can only imagine what the folks at Akamai were going through during the incident, along with everyone else affected. As we've seen repeatedly, however, it happens to all of us.

At Catchpoint, we're obsessed with user experience. Therefore, we want to know what happened, why, and what its impact was, so that we can learn from each other's mistakes and deliver better experiences on the Internet.

### The four pillars of Digital Experience Management

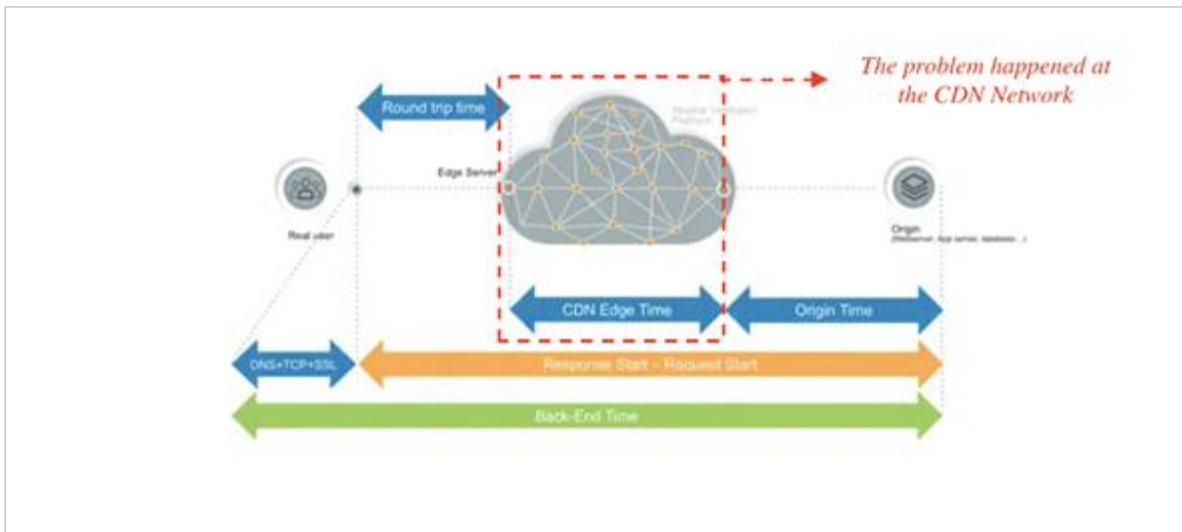
Performance is as important as the availability of any website. That's why at Catchpoint we believe in the following as the [four pillars of digital experience management](#):

- Reachability
- Availability
- Performance
- Reliability

### The three network components involved in using a CDN

When a customer is using a CDN, there are three network components involved:

- User to Edge, which represents the connectivity between end users and the CDN network.
- CDN Network, which represents the connectivity between edge servers within the CDN network.
- CDN Edge to origin, which represents the connectivity between the origin and CDN network.



The three network components in a CDN. (Catchpoint)

While static resources are usually cached and served from the edge, dynamic requests are routed through all three network components. Most monitoring solutions can only provide visibility into performance or availability issues for user to edge or CDN edge to origin. Catchpoint is the only active observability solution that enables businesses to detect performance issues within the CDN network itself.

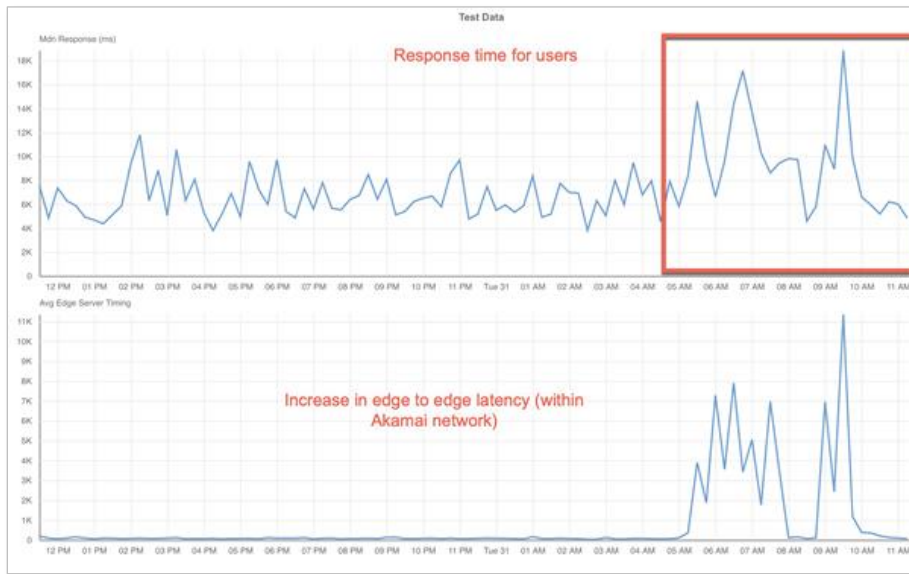
### Major websites hit by high response times

Starting at 10:00 UTC on August 31, we noticed that various websites around the world were having high response times. Sites affected included Expedia, the National Bank of Canada, AT&T, and Discover.

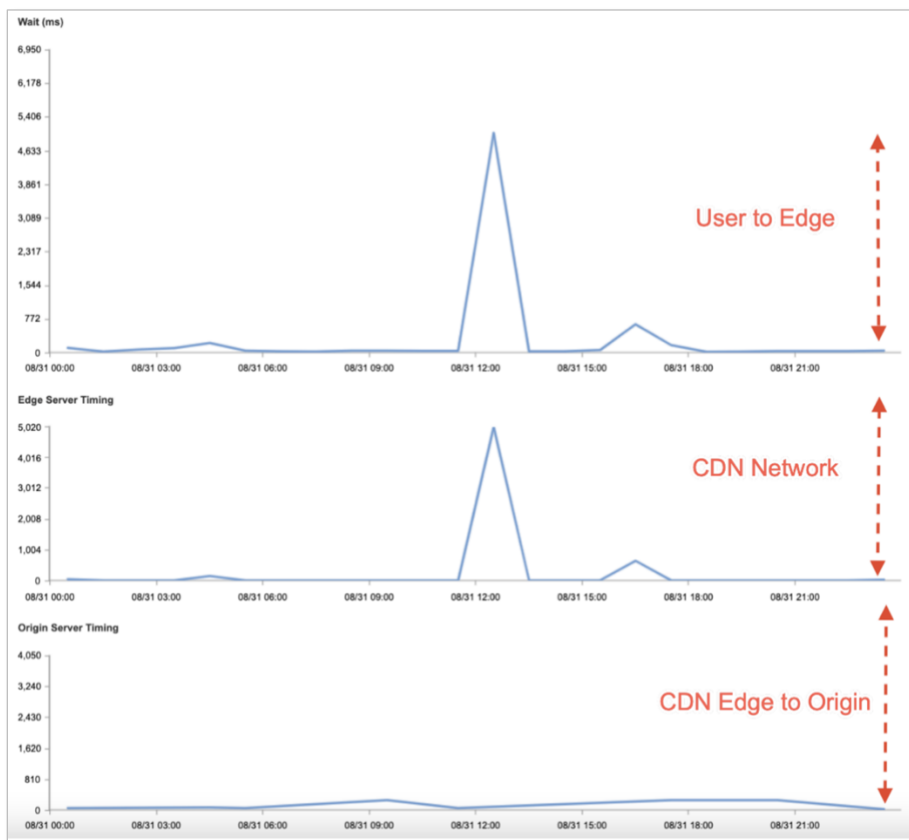
On further analysis, we found that the response time was directly impacted due to high wait times (i.e., Time to first Byte). High wait times commonly occur due to the following reasons:

- High server think/processing times.
- Issues in load balancers/gateways.
- Large backend processes.
- Server-side cache miss.
- Server resource utilization.

High wait times can also occur due to high latency within the CDN network or between the user and the edge.



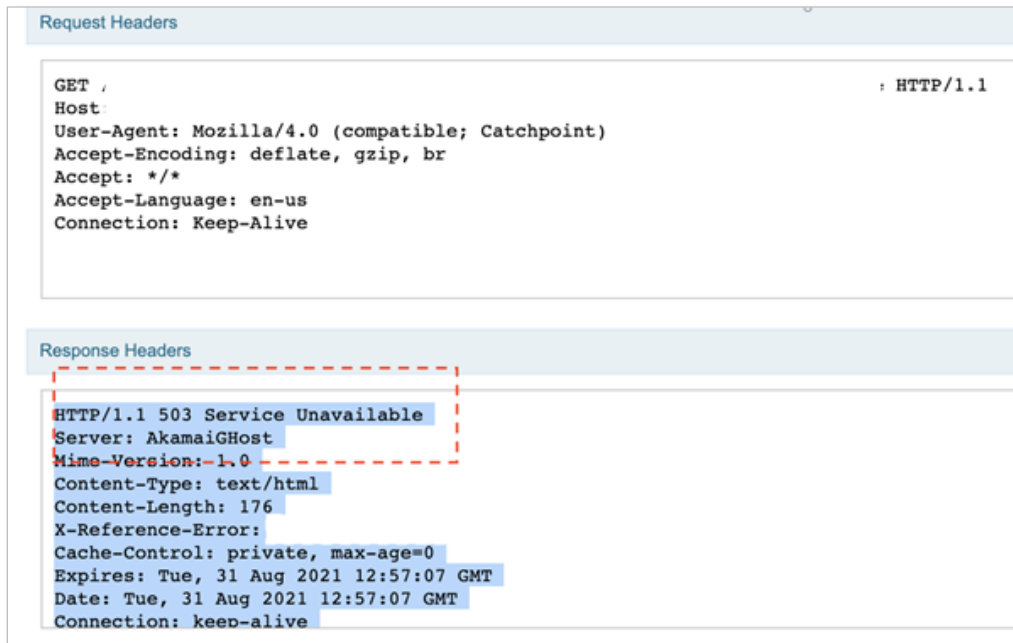
Performance graph showing how the increase in the CDN Network response time impacts response time. (Catchpoint)



Performance graph showing the issue specific to the CDN network while the origin did not have any problems. (Catchpoint)

## Availability drops due to 503 error codes

We also noticed availability drops as a result of 503 error codes from Akamai's ghost server. In this scenario, the request was trying to hit the origin server, but the request failed at the Akamai Edge network.



503 error code from Akamai edge server. (Akamai)

## Potential causes of latency in CDN networks

It is still not clear what was behind the performance issues at Akamai (it was rumored to be a wrong configuration deployment), but there are various potential causes for latency within a CDN network. These include:

- Routing between the CDN edge servers, which are managed by the CDN provider. If there are any routing issues between the edge servers, this could impact the time it takes to process incoming requests.
- The edge servers are overloaded or there is a surge in inbound traffic.
- There is a wrong CDN configuration.
- Issues with third-party solutions utilized by the CDN.



## Be prepared to act quickly

Businesses use a CDN to be as close as possible to the end user so they can be faster, no matter their geography. To find out if your CDN is impacting your users negatively, you must monitor from closer to where your users are – so that you can identify and act on any performance issues such as this one as quickly as possible.

*Published on Sep 02, 2021*

---

*"Content delivery networks (CDNs) are vital to delivery of Internet services, as they front-end almost all web sites, delivering their content to users quickly through local proxies. They rely on fast resolution of domain name service (DNS) requests and are vulnerable to misconfiguration and errors in their DNS systems.*

*Because CDNs are invisible to end users, they can be overlooked in corporate resilience planning. The event underlines the need to avoid having a single point of failure."*

*~Peter Judge, Global Editor, Datacenter Dynamics*

---

## Incident Review: June 8, 2021 – Fastly Outage Impacts Major Websites Worldwide

By Kameerath Kareem

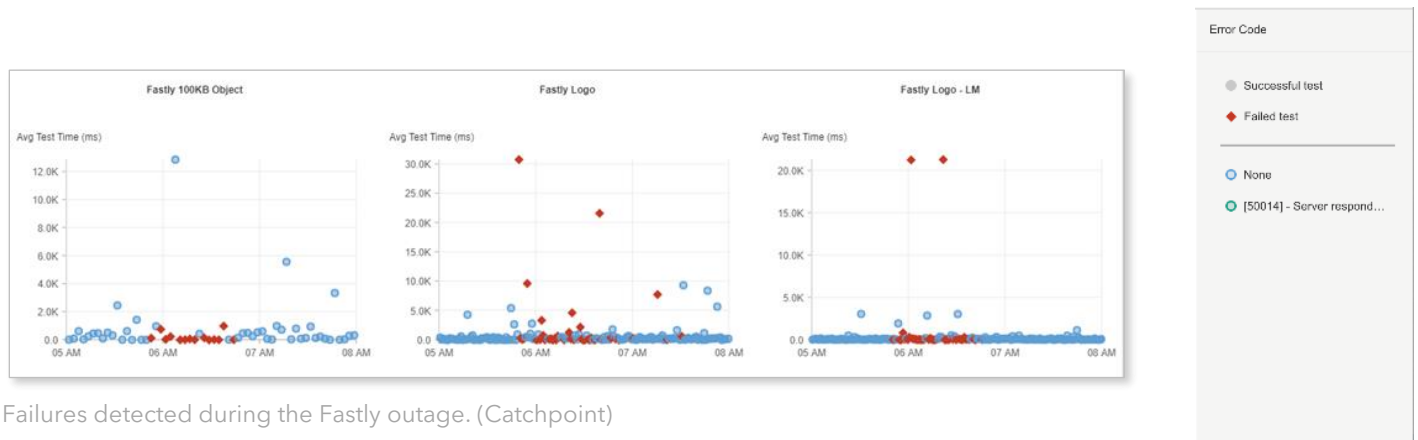
On June 8, 2021, many of us were left staring at blank screens or “Service Unavailable” errors when trying to access the Internet. This panic was shared by millions of people around the world. Major websites around the world from Amazon to Vimeo were inaccessible to users. Why? A major outage to leading CDN provider Fastly impacted anyone using their services.

Here is a quick rundown of what happened and why.

### Internet outage timeline

The issues began at 09:50 UTC on June 8 and lasted till 10:45 UTC. The hour-long outage impacted thousands of websites, applications, and services that relied on Fastly due to the CDN provider suffering an outage.

The scatterplot below shows that the content from Fastly was unavailable globally due to 5xx errors that started at around 9:50 UTC.



Failures detected during the Fastly outage. (Catchpoint)

Countless popular sites, including Twitch, Stack overflow, Spotify, Pinterest, HBO Max, Hulu, Shopify, PayPal, Reddit, GitHub, HBO, Amazon, and many more that were hosted on Fastly were impacted.

Below is a look at some of the websites that suffered from the outage. Once the fix was applied by Fastly, these sites were back up and running. However, we do see that it took some time for the sites to be up completely due to errors cached at the time of the outage.

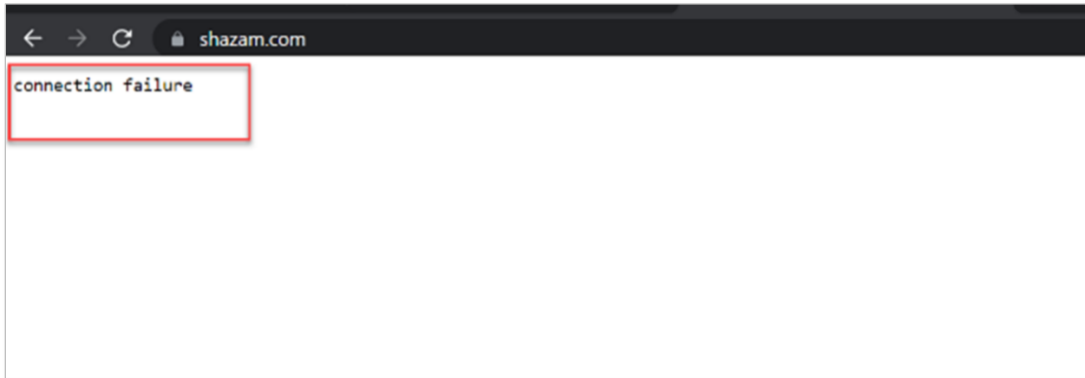


Websites impacted by the Fastly outage. (Catchpoint)

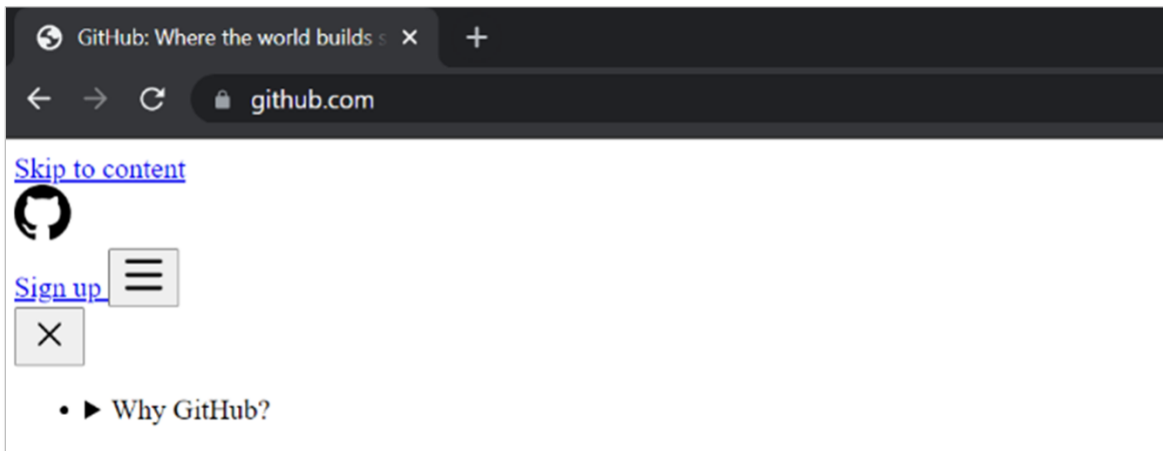
The end users were seeing 503 Service Unavailable and 502 Connection Failures. Some users saw a broken page since cached static content wasn't loading as expected.



Error message. (web.dev)



Connection failure message (Shazam.com)



Broken page (Github.com)

*"Different websites handled the outage in different ways. The Guardian moved to Twitter to run a dedicated liveblog, while tech news site the Verge published news to a shared Google Doc - until a reporter accidentally shared a link on Twitter that allowed the audience to edit it.*

*The increasing centralization of internet infrastructure in the hands of a few large companies means that single points of failure can result in sweeping outages. In 2017, a problem at Amazon's AWS hosting business, for instance, took out some of the world's biggest websites for several hours across the entire US east coast."*

*~Alex Hern, Technology Editor, The Guardian*

## Fastly acknowledges the problem

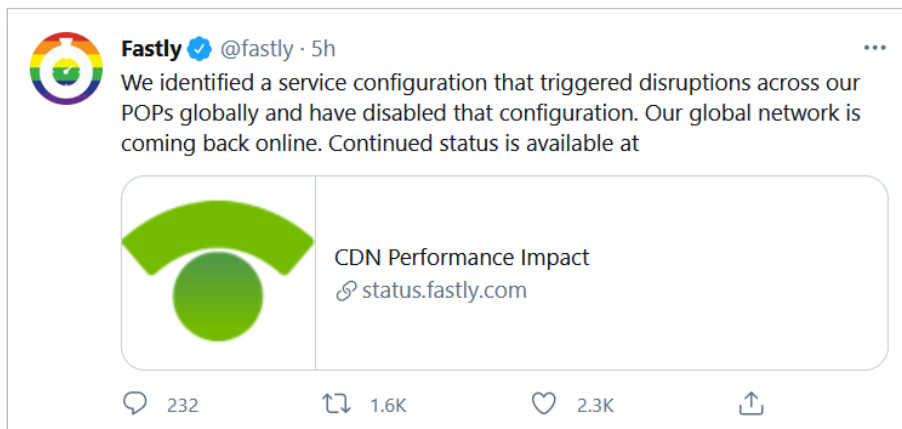
Fastly updated their [status page](#) in a timely manner, acknowledging the impact on websites across the globe.

⊞ North America	Degraded Performance
⊞ South America	Degraded Performance
⊞ Europe	Degraded Performance
⊞ Asia/Pacific	Degraded Performance
⊞ South Africa	Degraded Performance
⊞ India	Degraded Performance

Fastly status page during outage. (Catchpoint)

Clearly, transparency was important to Fastly during the outage since they constantly refreshed the page to ensure their customers were aware of what was being done and to tell them when they could expect systems to be fully restored.

During the outage, Fastly communicated that it had identified, “a service configuration that triggered disruptions” across its servers around the world. Once they had pinpointed the cause, the CDN disabled that configuration to resolve the issue.



Fastly Tweet sharing status during outage. (Twitter/@fastly)

The impacted sites recovered once Fastly fixed the configuration issue.



Site performance after the issue was resolved. (Catchpoint)

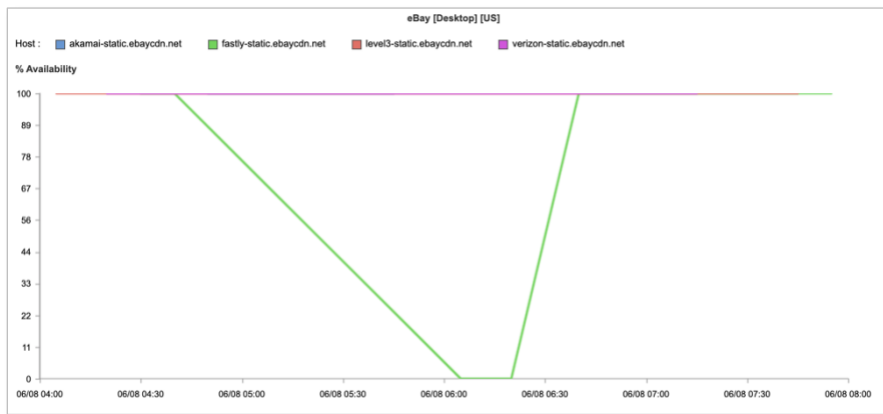
It was [revealed afterwards](#) that the Internet blackout was caused by one customer updating their settings, which triggered a bug. According to Nick Rockwell, the company's head of engineering and infrastructure, a bug in Fastly's code introduced in mid-May had lain dormant until the time of the incident. The updating of settings triggered the bug, which led to 85% of the Fastly network to return errors.

## Why employ a multi-CDN strategy?

Fastly was able to quickly rectify the situation while keeping their customers updated. However, many users had already been impacted and suffered the consequences.

This incident is another in a long list of CDN-related outages that can be swiftly managed with a multi-CDN strategy. Failover options are a must, to avoid situations like this where your website is rendered inaccessible to users. Rerouting traffic to origin servers may help temporarily, but at the cost of performance. It would be better to switch CDNs to mitigate the impact of the outage on the user.

When Fastly went down, eBay had performance issues, but did not suffer a complete outage due to having a multi-CDN strategy in place.



eBay site availability over different CDNs. (Catchpoint)

## Conclusions and advice

Any multi-CDN implementation would be incomplete without a CDN observability strategy. It is vital to constantly monitor CDN providers to catch such outages early on.

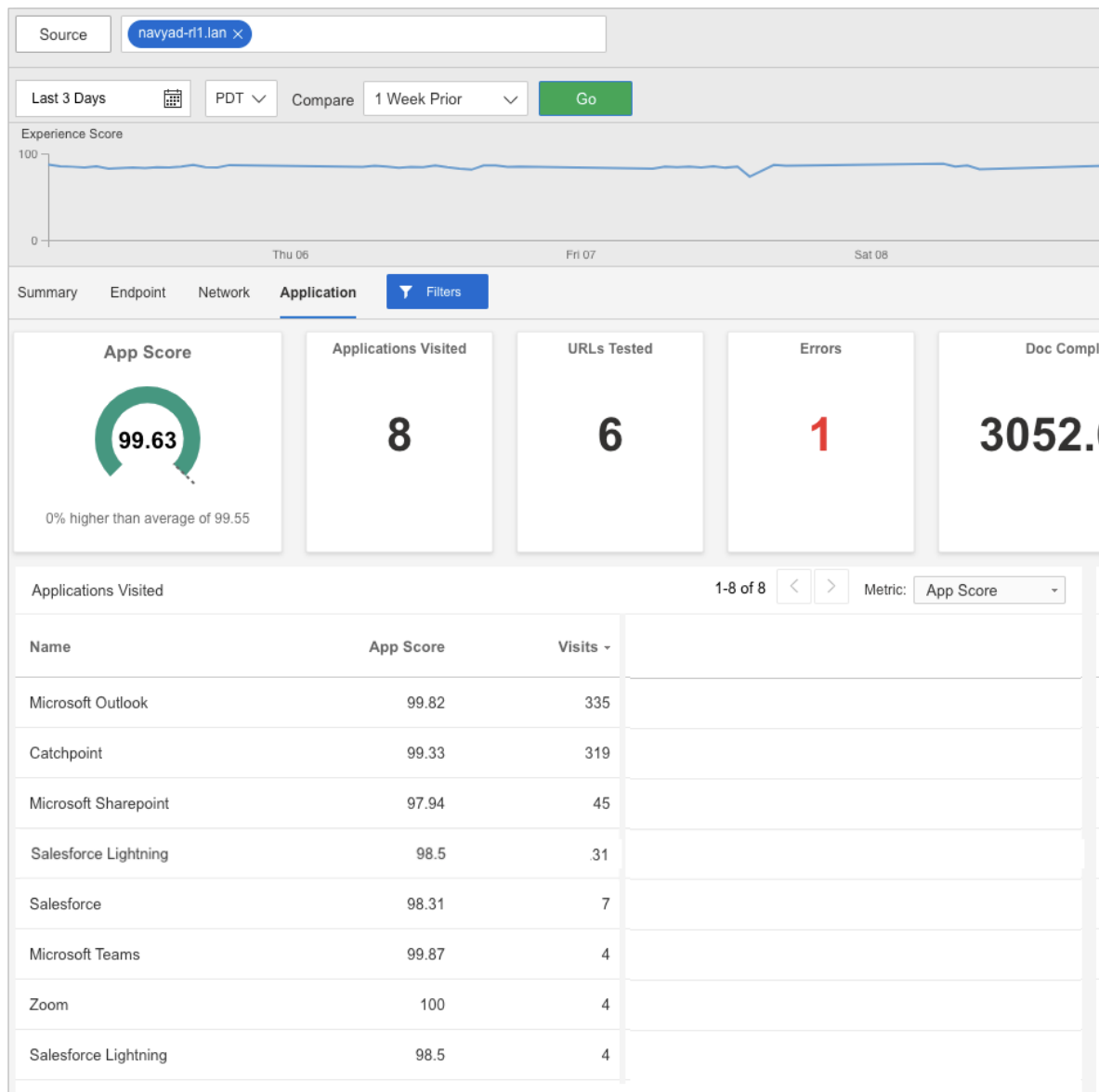
Moreover, the incident is a reminder that implementing any change without monitoring the impact of that change can be detrimental to your business. Proactively monitoring your vendors gives you a head start, so you can quickly deploy failover strategies to mitigate any resulting impact on end user experience.

*Published on Jun 08, 2021*

## Incident Review: May 11, 2021 - Anatomy of The Salesforce Outage

By Zachary Henderson

At Catchpoint, I work as a Solutions Engineer. Being on the sales side, one of the applications I use a lot is Salesforce, the CRM platform used at Catchpoint and thousands of other organizations. According to Catchpoint's Endpoint data, Salesforce is my fourth most visited site.



Per User Application Scores and Traffic Volumes. (Catchpoint)



When Salesforce had an outage for close to five hours yesterday, the first thought that came to mind was, "Thank God it's not quarter-end or worse year-end." Quarter-end is when sales teams are hustling to close deals. Year-end, the last quarter, is when every deal matters as it impacts the revenue of the company. Salesforce is critical for sales teams during this time and its availability and performance directly impact not only employee productivity but also business results.

Since efficiency is key in any organization, our sales processes are automated using Salesforce. We rely on it for requesting a service order, routing contracts for signature, for getting billing information, and so on. So yes, a Salesforce outage can result in missing the quarterly targets!

### "We drink our own champagne at Catchpoint"

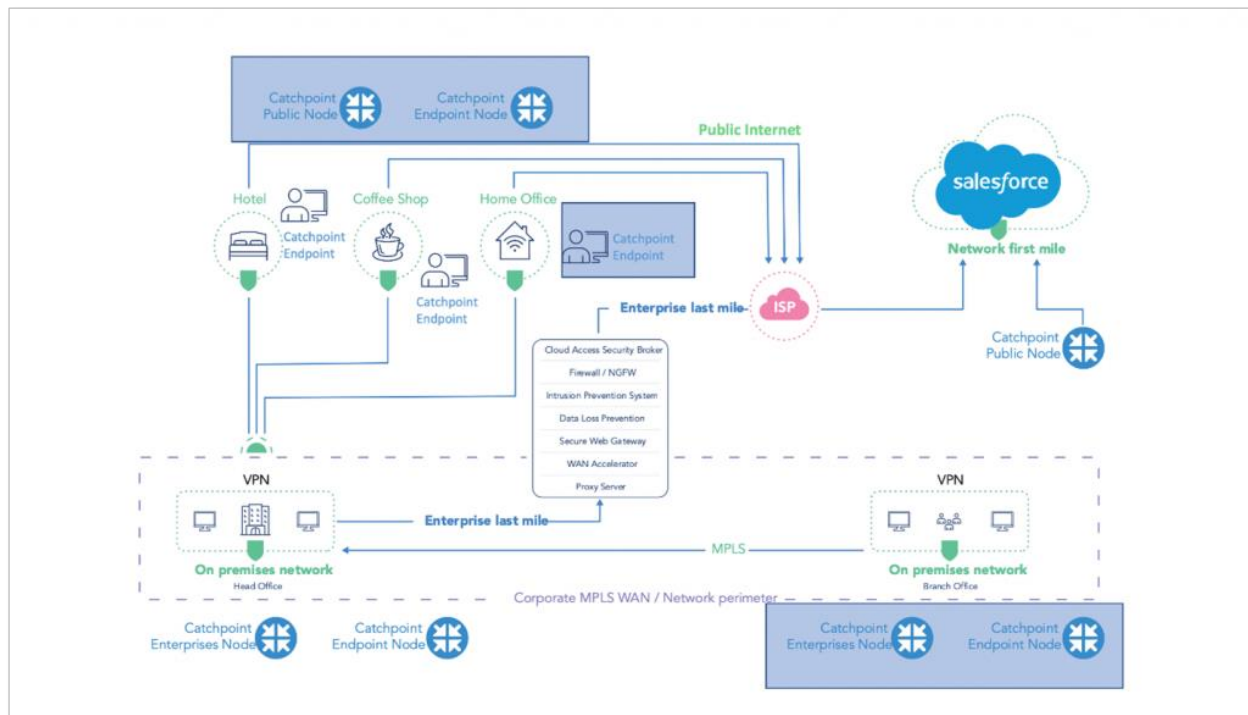
Working for a monitoring company also means I have the fortune of getting notified about any issue that impacts our systems and applications before it impacts users. We leverage Active, Real User, Endpoint, and Network monitoring to monitor our backend systems, our customer-facing portal, our APIs and web services, and every SaaS application used in the organization.

As our CEO Mehdi puts it, "We drink our own champagne at Catchpoint."

Full visibility and performance control of an application is best achieved by leveraging multiple data sources, giving you complementary perspectives.

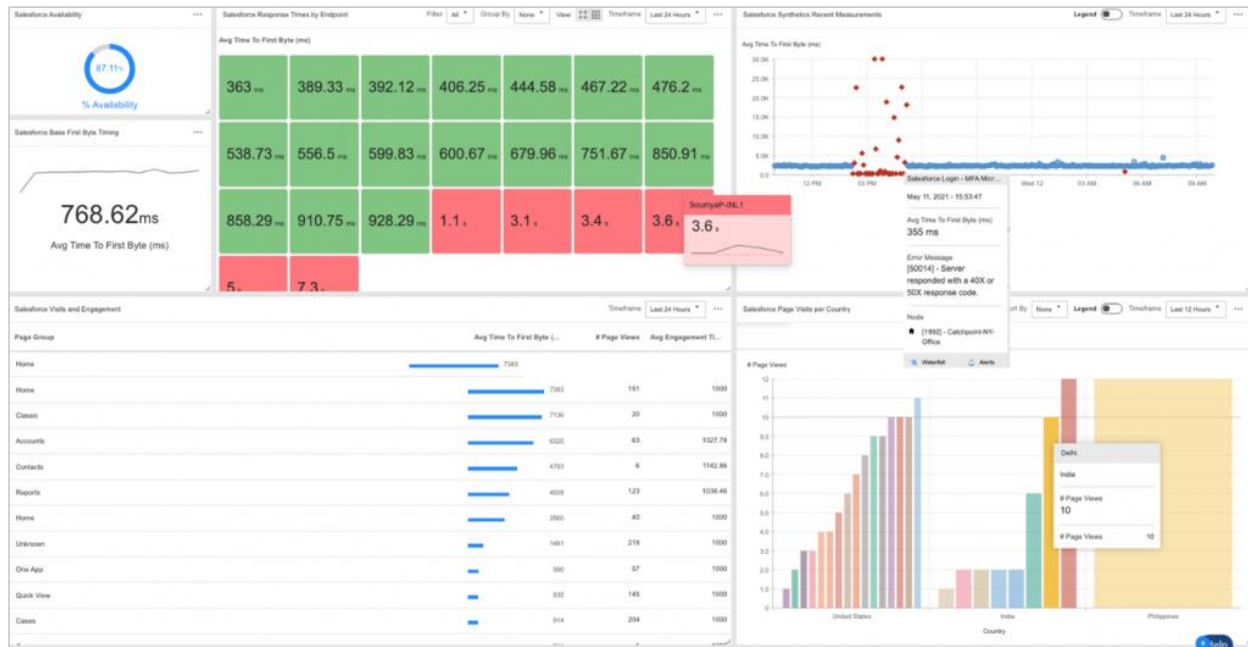
### Our monitoring strategy for Salesforce

At a high level, our monitoring strategy for Salesforce is summarized in the diagram below.



Monitoring viewpoints in relation to Salesforce and user locations. (Catchpoint)

Our overview dashboard for Salesforce combines various data sources to give us a holistic view into Salesforce availability and performance.



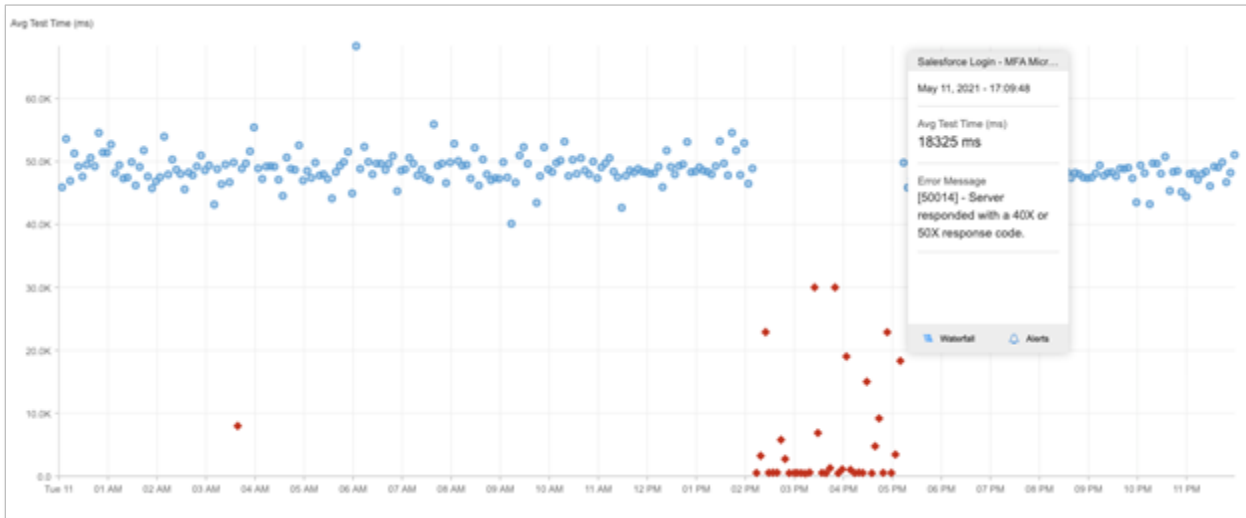
Salesforce overview dashboard highlighting degraded availability and user availability down to specific time window and end users. (Salesforce)

The dashboard shows that Salesforce was hard down between 21:03 UTC on May 11, 2021, until 02:19 UTC on May 12, 2021.

Let me take you through the perspectives we get from combining proactive and reactive monitoring.

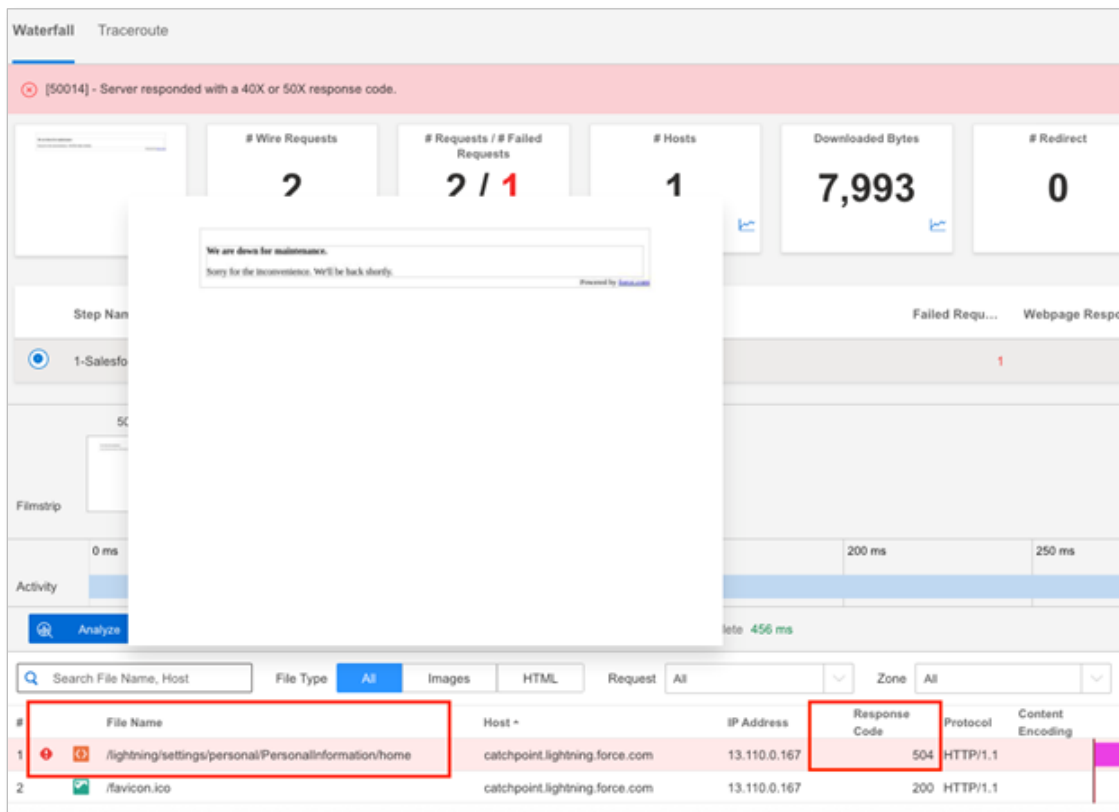
**Active Monitoring from Catchpoint Public and Enterprise Observers** – We use these to proactively monitor end-to-end user journeys. Before the pandemic, we only used on-prem observers which are deployed in our branch offices. Post-COVID, we added monitoring from backbone and broadband observers since employees are working from home and accessing SaaS applications through the public Internet. This is the observability capability that tells us there is an issue even before our IT teams see a ticket from an employee.

Below is the active observability data for a critical user journey for Salesforce from yesterday. The red diamonds show the outage.



Timeframe of recent Salesforce measurements, the red diamonds are the Salesforce outage. (Catchpoint)

We were able to drill down to know exactly what was going on.



HTTP 504 error on the Salesforce landing page. (Catchpoint)

The landing page after you login, Salesforce Home, failed to load, with the server returning an HTTP 504 error. Users saw the maintenance page.

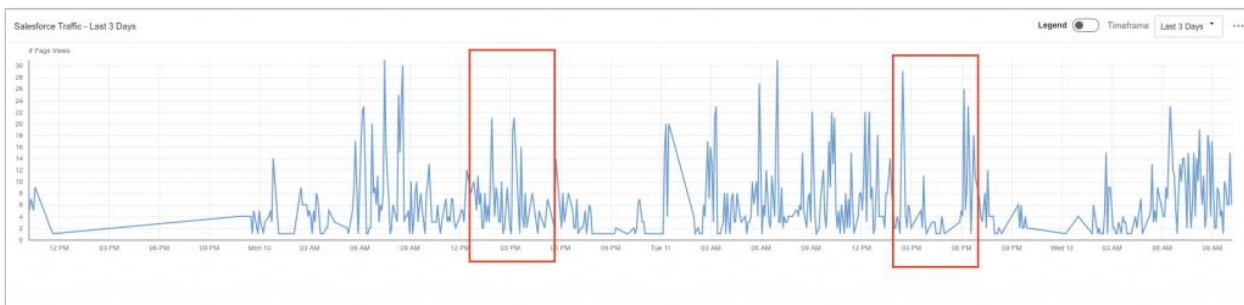
**Endpoint Observability** – Every employee at Catchpoint has the Catchpoint endpoint agent deployed on their laptops. This gives us real user data on the performance and availability of every SaaS application from the real user perspective. However, we love the power of active observability and also run proactive tests for critical applications from the Endpoint agents.

During the outage, we saw a dip in page visits to Salesforce since the application was down at the landing page itself.



Data showing dip in page visits to Salesforce. (Catchpoint)

Looking at data from the last three days, the time period of the Salesforce outage is also a time when the application is used a great deal. You can see the surge below in the page views after the outages – looks like several of my colleagues had to put in some late hours to wrap up work yesterday!



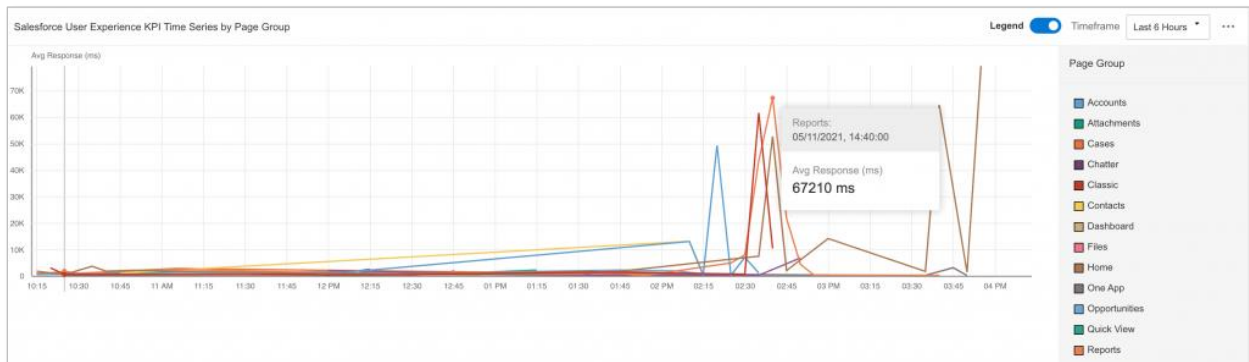
Data showing traffic during the peak hours of the Salesforce outage. (Catchpoint)

It doesn't stop at that. With Endpoint Observability, we can also answer further questions such as:

- Who was impacted?
- Where were they accessing the application from?
- Is it their network or is it the application?
- What parts of the application are impacted?

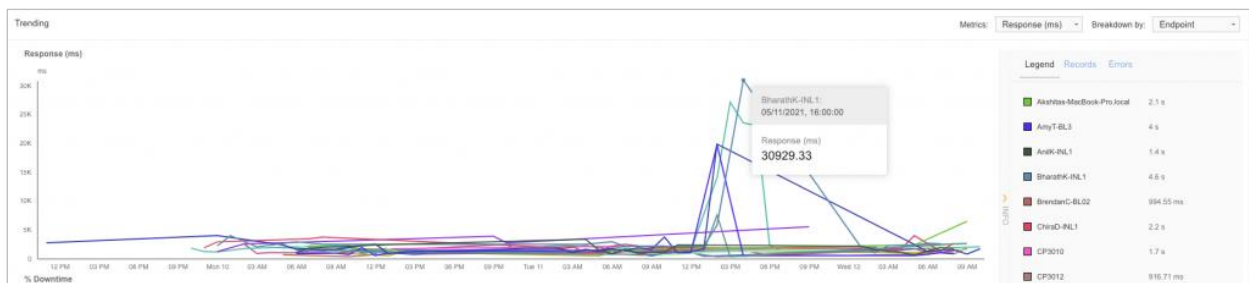
Sometimes only parts of an application are slow or down and having this level of granularity helps IT teams better assist employees when there is an issue.

Endpoint data below shows employees were seeing high response time and errors accessing all parts of the Salesforce application.



High response times across the major Salesforce app components as measured from real user traffic. (Catchpoint)

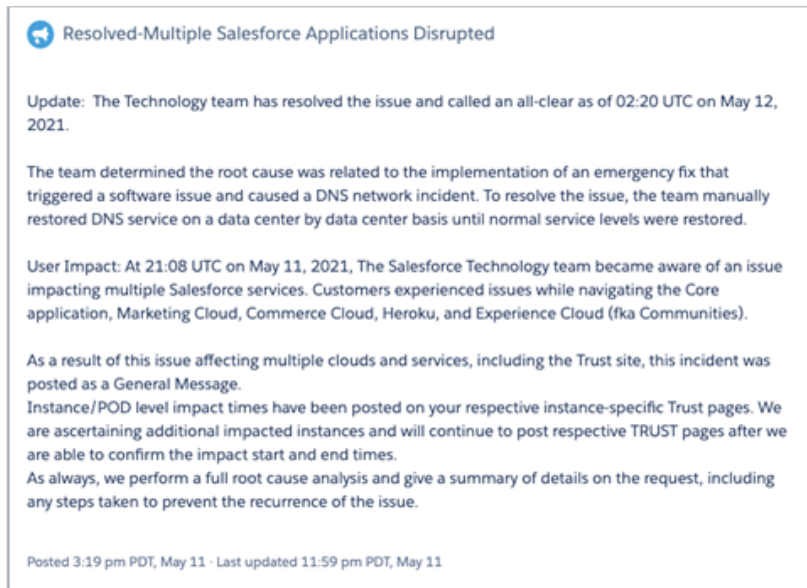
From the screenshot below, we can see exactly who had the worst impact.



Individual users who were impacted by the Salesforce outage. (Catchpoint)

## The Salesforce summary of the outage

Outages are difficult. Thank you to the Salesforce team for working hard to resolve the issue. Below is the summary of the outage that was posted on the Salesforce status page (which itself was down for a portion of the outage).



Summary of the outage from Salesforce outage page. (Salesforce)

## Developing a sound monitoring and observability strategy for SaaS

As organizations are adopting cloud, utilizing microservices architectures, and increasing the use of SaaS applications, traditional forms of monitoring using APM, NPM, logging, and tracing fall short. How can you instrument code and applications you don't own?

---

*...(L)urking within that tried-and-trusted script was a bug. Under load, a timeout could happen that would stop other things from running. And sure enough, as the update was being rolled out across all of Salesforce's data centers, a timeout occurred. This in turn meant that certain tasks were not carried out when the servers were restarted. And that, in turn, meant that those servers did not return to operation correctly. That left customers unable to access Salesforce's products.*

*~Richard Speed, Writer, The Register*

---

*Published on May 12, 2021*

## Incident Review: May 6, 2021 – How Catchpoint Resolved Issues Caused by the Neustar UltraDNS Outage

By Varun Master

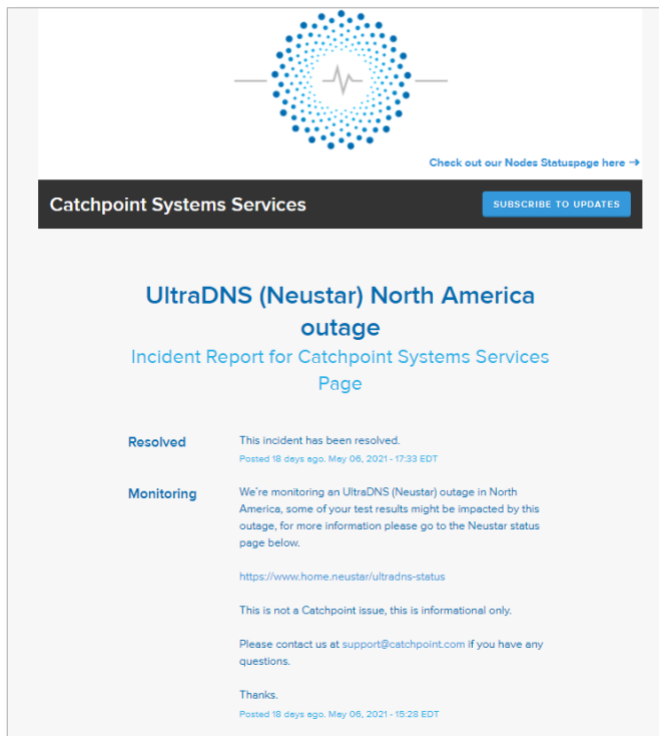
At Catchpoint, [our award-winning support team](#) aims to be a partner, not just a gateway to the tool. When UltraDNS, a major DNS provider, went down on May 6, 2021, we found ourselves faced with nine support tickets within one hour.

Our customers were experiencing outages on their websites and online services. They needed urgent help from our team to understand what was causing the disruption, so they could quickly resolve the situation or validate their own findings that it was a third-party DNS issue.

### What happened with UltraDNS?

Many of our clients use external DNS providers, UltraDNS is one of them. The outages started occurring on May 6 at around 17:45 UTC on the U.S. East coast and in the U.S. Central and U.S. West regions.

Many companies use Catchpoint to observe their digital services, such as websites and API services, to ensure they're running as quickly as possible. These businesses started to get alerts from Catchpoint about a large number of DNS failures. Our support team quickly swung into action to help our customers.



The screenshot shows a web page titled "UltraDNS (Neustar) North America outage" with the subtitle "Incident Report for Catchpoint Systems Services Page". At the top, there is a logo consisting of a circle of blue dots with a white heartbeat line in the center, and a link "Check out our Nodes Statuspage here ->". Below the logo is a dark banner with "Catchpoint Systems Services" and a "SUBSCRIBE TO UPDATES" button. The main content area has a "Resolved" status with the text "This incident has been resolved." and a timestamp "Posted 18 days ago, May 06, 2021 - 17:33 EDT". Below this is a "Monitoring" section with the text "We're monitoring an UltraDNS (Neustar) outage in North America, some of your test results might be impacted by this outage, for more information please go to the Neustar status page below." and a link "https://www.home.neustar/ultradns-status". It also includes a disclaimer "This is not a Catchpoint issue, this is informational only.", contact information "Please contact us at support@catchpoint.com if you have any questions.", and a "Thanks." message with a timestamp "Posted 18 days ago, May 06, 2021 - 15:28 EDT".

Catchpoint incident report for UltraDNS. (Catchpoint)

We already knew what was going on, since we were monitoring all the major infrastructure providers, including UltraDNS. This was a real outage, happening worldwide with Neustar UltraDNS. We immediately [posted information on our status page](#) to inform all our customers and keep them updated as the situation unfolded.

### **What did the velocity and volume of the tickets that were coming in look like?**

Although we had updated our status page and the Catchpoint alerts clearly showed the error types, nine customer tickets still came in within the hour. DNS outages like this don't happen too often. They can be a significant problem for customers, particularly those with a single DNS provider.

The whole ordeal lasted for about an hour.

### **What did clients experience?**

The first ticket came from the Director of Software Development at a cybersecurity firm. His first question was whether we rolled out any changes over the last few hours. He confirmed they were seeing "errors globally" and wanted to find out what the root cause was.

Our agent joined the live chat within a few seconds and sought to clarify what was going on. The firm let our agent know that they were also talking to their DNS provider (UltraDNS), who then confirmed that they were indeed having issues.

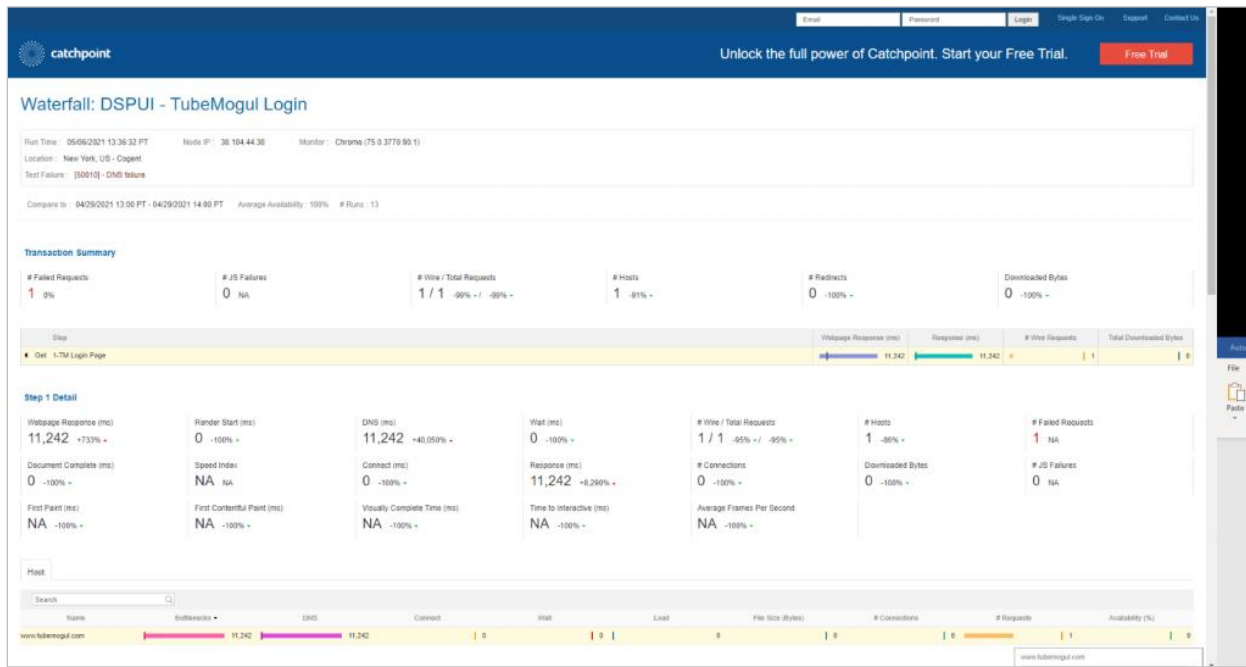
A leading Software as a Service (SaaS) company was the next client to reach out. Their Operations staff asked if there was "something going on here we should be aware of with the observers" or if it was "a bigger network issue."

They shared the affected test link, which showed DNS failures in New York, New York, U.S. and Washington DC, U.S.

Because we never take anything for granted, our agent said they would need some time to research what was going on and created a ticket. We never assume that because vendor ABC is having issues, all the alerts we are seeing are due to that vendor.

In this case, because the issue was related to a third party, we helped our customers understand what was going on, what the root cause of the issue was, and how they could work with the provider to continue to stay informed about the situation. We also provided alternative strategies and kept them informed as the situation progressed.



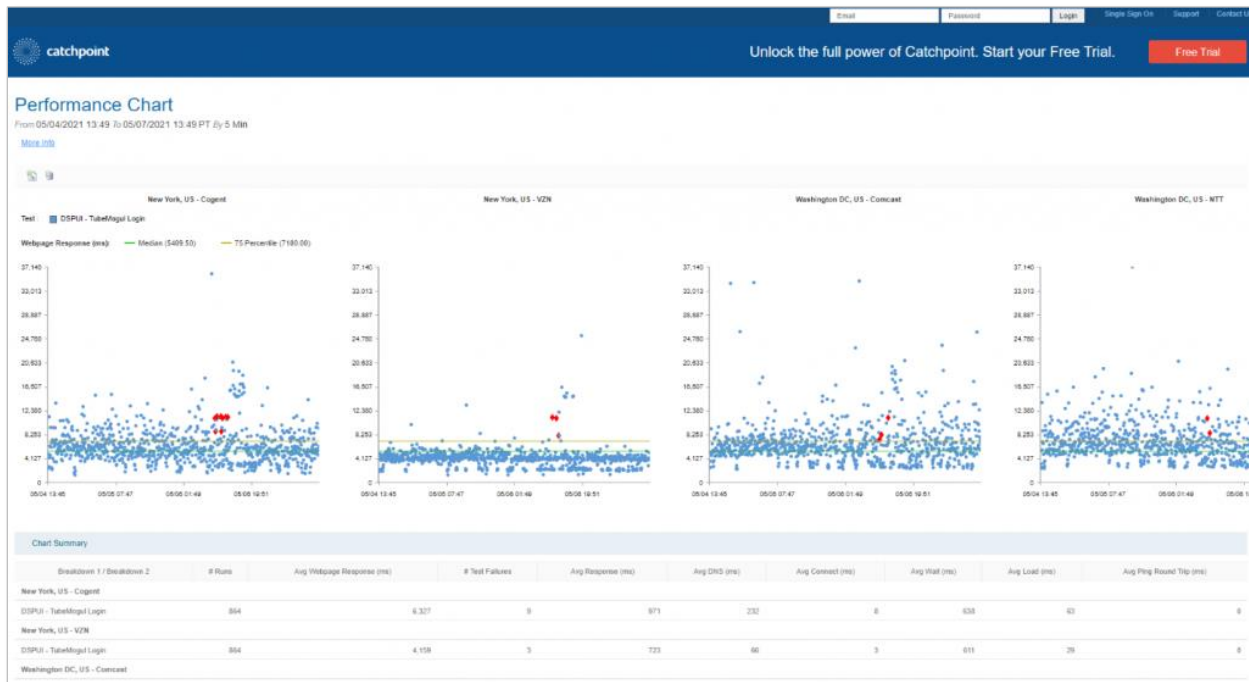


Client test run showing a spike in DNS metric taking around 11 seconds and test failing with DNS Failure. (Catchpoint)

## How we ensure support relationships are true partnerships

At every stage of the incident response, we seek to partner with our customers and help carry some of their load. We remove stress by providing visibility into the network.

One of the ways that we did this during the UltraDNS outage was to monitor their tests for several hours after the situation was resolved. We also sent links to the tests in Catchpoint that we manage. These tests provide data beyond what the average customer would have, showing that their DNS services are now running as expected.



All test runs showing test failure across all nodes for the duration of the DNS outage. (Catchpoint)

## An ideal customer support response to an Internet outage

At Catchpoint, we know that issues can have wide and varied sources, as well as deep impacts on our customers. That's why we always respond immediately to any ticket that comes in. Here is our process, and what we consider to be best practices for any observability solution:

First, ensure our customers understand the issue. We understand what customers go through when their websites or services are down, and our support team is on standby to help resolve the issue.

Next, run an investigation to find out what's going on.

As soon as the source of the issue is discovered, relay that information to our customers.

If the issue is related to a third-party service (like the one described above), inform our customers of the root cause of the issue and work with the provider to ensure everyone stays abreast of the situation's status.

Offer alternatives. For example, we might suggest customers pause alerts until things go back to normal.

Finally, once we have additional details to share, do so immediately.

*Published on May 27, 2021*

## Conclusion

Website downtime can happen to anyone at any time. The costs to business and reputation can be profound.

In a complex digital landscape, which is increasingly reliant on the cloud and Internet for businesses to function, you need to be prepared for any kind of downtime or latency. It doesn't matter if the issue is caused by a bug, a misconfiguration, your cloud provider, an ISP going down, or an issue with one of your providers – if you don't know the source of the problem and what to do next, you'll be hard pressed to minimize damage to your business.

Deep visibility into the full digital experience is essential. With a comprehensive digital observability solution, you can take proactive steps to correct issues as quickly as possible, in the moment, and improve your response in the future.

As you develop and improve your observability strategy, ensure your visibility perspective allows you to gather as much data as possible: from the networks, services, and every online service an employee or customer touches during a transaction.

In this way, you can respond quickly to issues and be proactive about communication to your own customers. In addition, you can verify that your providers are living up to the SLAs you've contracted. If not, you can demand restitution, switch providers, or add better failover options.

---

**Stay up to date on the most recent outages**

[Subscribe to our blog](#)

---

### About Catchpoint

Catchpoint is the enterprise-proven Digital Experience Observability industry leader, empowering teams to own the end user experience. Because we provide unparalleled visibility and insight, Fortune 500 enterprises trust Catchpoint's observability platform to proactively and rapidly detect and repair problems before they impact digital user experience. Learn more at [www.catchpoint.com](http://www.catchpoint.com).