



# Cloudflare outage on November 18, 2025

2025-11-18



Matthew Prince

12 min read

This post is also available in [English](#), [Français](#), [Deutsch](#), [日本語](#), [한국어](#), [Español](#), [Nederlands](#) and [हिन्दी](#).



On 18 November 2025 at 11:20 UTC (all times in this blog are UTC), Cloudflare's network began experiencing significant failures to deliver core network traffic. This showed up to Internet users trying to access our customers' sites as an error page indicating a failure within Cloudflare's network.

# Internal server error

Error code 500

Visit [cloudflare.com](https://cloudflare.com) for more information.

2025-11-18 11:35:32 UTC



You  
Browser  
Working



London  
Cloudflare  
Error



Washington, DC  
Host  
Working

## What happened?

There is an internal server error on Cloudflare's network.

## What can I do?

Please try again in a few minutes.

Cloudflare Ray ID: 9a072a3bce88467f • Your IP: [Click to reveal](#) • Performance & security by [Cloudflare](#)

**The issue was not caused, directly or indirectly, by a cyber attack or malicious activity of any kind.** Instead, it was triggered by a change to one of our database systems' permissions which caused the database to output multiple entries into a “feature file” used by our Bot Management system. That feature file, in turn, doubled in size. The larger-than-expected feature file was then propagated to all the machines that make up our network.

The software running on these machines to route traffic across our network reads this feature file to keep our Bot Management system up to date with ever changing threats. The software had a limit on the size of the feature file that was below its doubled size. That caused the software to fail.

After we initially wrongly suspected the symptoms we were seeing were caused by a hyper-scale DDoS attack, we correctly identified the core issue and were able to stop the propagation of the larger-than-expected feature file and replace it with an earlier version of the file. Core traffic was largely flowing as normal by 14:30. We worked over the next few hours to mitigate increased load on various

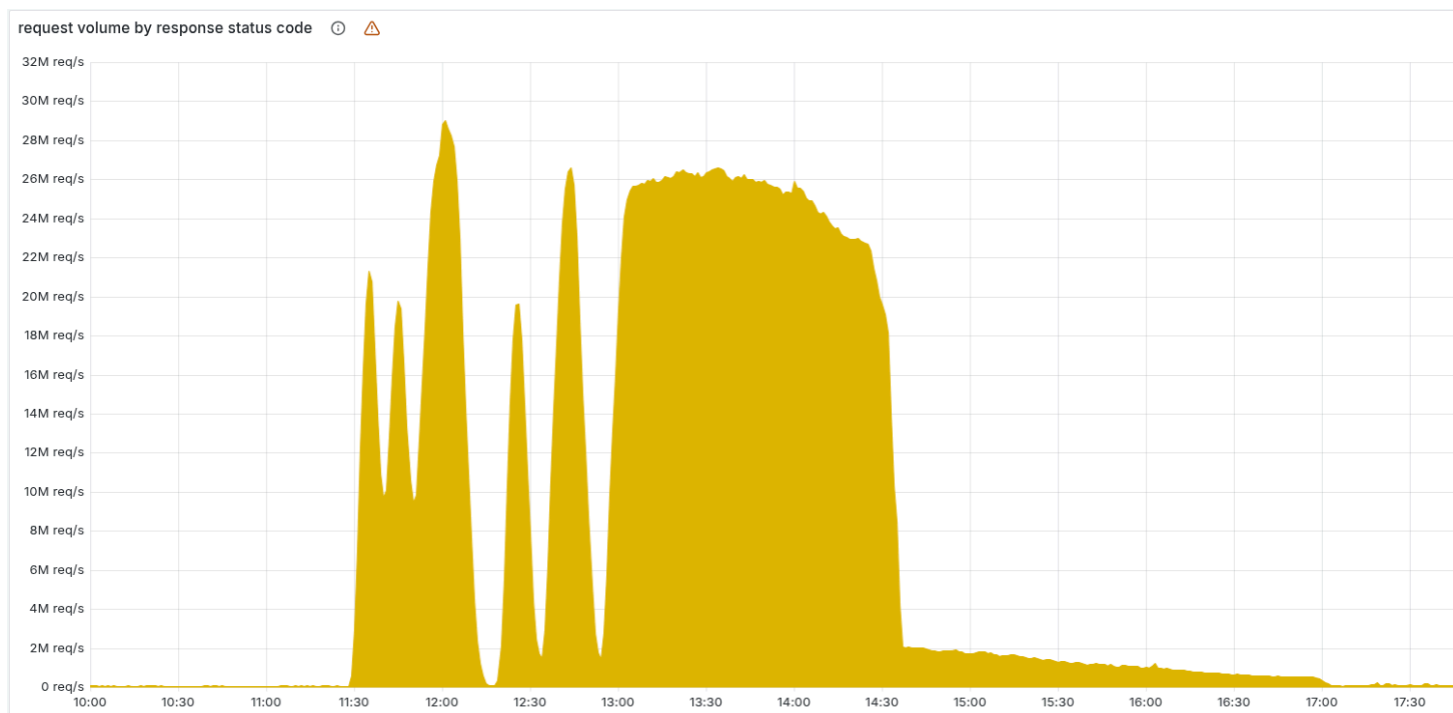
parts of our network as traffic rushed back online. As of 17:06 all systems at Cloudflare were functioning as normal.

We are sorry for the impact to our customers and to the Internet in general. Given Cloudflare's importance in the Internet ecosystem any outage of any of our systems is unacceptable. That there was a period of time where our network was not able to route traffic is deeply painful to every member of our team. We know we let you down today.

This post is an in-depth recount of exactly what happened and what systems and processes failed. It is also the beginning, though not the end, of what we plan to do in order to make sure an outage like this will not happen again.

## The outage [↗](#)

The chart below shows the volume of 5xx error HTTP status codes served by the Cloudflare network. Normally this should be very low, and it was right up until the start of the outage.



The volume prior to 11:20 is the expected baseline of 5xx errors observed across our network. The spike, and subsequent fluctuations, show our system failing due to loading the incorrect feature file. What's notable is that our system would then recover for a period. This was very unusual behavior for an internal error.

The explanation was that the file was being generated every five minutes by a query running on a ClickHouse database cluster, which was being gradually updated to improve permissions management. Bad data was only generated if the query ran on a part of the cluster which had been updated. As a result, every five minutes there was a chance of either a good or a bad set of configuration files being generated and rapidly propagated across the network.

This fluctuation made it unclear what was happening as the entire system would recover and then fail again as sometimes good, sometimes bad configuration files were distributed to our network. Initially, this led us to believe this might be caused by an attack. Eventually, every ClickHouse node was generating the bad configuration file and the fluctuation stabilized in the failing state.

Errors continued until the underlying issue was identified and resolved starting at 14:30. We solved the problem by stopping the generation and propagation of the bad feature file and manually inserting a known good file into the feature file distribution queue. And then forcing a restart of our core proxy.

The remaining long tail in the chart above is our team restarting remaining services that had entered a bad state, with 5xx error code volume returning to normal at 17:06.

The following services were impacted:

Service / Product	Impact description
Core CDN and security services	HTTP 5xx status codes. The screenshot at the top of this post shows a typical error page delivered to end users.
Turnstile	Turnstile failed to load.
Workers KV	Workers KV returned a significantly elevated level of HTTP 5xx errors as requests to KV’s “front end” gateway failed due to the core proxy failing.

Dashboard	While the dashboard was mostly operational, most users were unable to log in due to Turnstile being unavailable on the login page.
Email Security	While email processing and delivery were unaffected, we observed a temporary loss of access to an IP reputation source which reduced spam-detection accuracy and prevented some new-domain-age detections from triggering, with no critical customer impact observed. We also saw failures in some Auto Move actions; all affected messages have been reviewed and remediated.
Access	<p>Authentication failures were widespread for most users, beginning at the start of the incident and continuing until the rollback was initiated at 13:05. Any existing Access sessions were unaffected.</p> <p>All failed authentication attempts resulted in an error page, meaning none of these users ever reached the target application while authentication was failing. Successful logins during this period were correctly logged during this incident.</p> <p>Any Access configuration updates attempted at that time would have either failed outright or propagated very slowly. All configuration updates are now recovered.</p>

As well as returning HTTP 5xx errors, we observed significant increases in latency of responses from our CDN during the impact period. This was due to large amounts of CPU being consumed by our debugging and observability systems, which automatically enhance uncaught errors with additional debugging information.

## How Cloudflare processes requests, and how this went wrong today

Every request to Cloudflare takes a well-defined path through our network. It could be from a browser loading a webpage, a mobile app calling an API, or automated traffic from another service. These requests first terminate at our HTTP and TLS layer, then flow into our core proxy system (which we call FL for “Frontline”), and finally through Pingora, which performs cache lookups or fetches data from the origin if needed.

We previously shared more detail about how the core proxy works [here](#).

As a request transits the core proxy, we run the various security and performance products available in our network. The proxy applies each customer's unique configuration and settings, from enforcing WAF rules and DDoS protection to routing traffic to the Developer Platform and R2. It accomplishes this through a set of domain-specific modules that apply the configuration and policy rules to traffic transiting our proxy.

One of those modules, Bot Management, was the source of today's outage.

Cloudflare's [Bot Management](#) includes, among other systems, a machine learning model that we use to generate bot scores for every request traversing our network. Our customers use bot scores to control which bots are allowed to access their sites — or not.

The model takes as input a “feature” configuration file. A feature, in this context, is an individual trait used by the machine learning model to make a prediction about whether the request was automated or not. The feature configuration file is a collection of individual features.

This feature file is refreshed every few minutes and published to our entire network and allows us to react to variations in traffic flows across the Internet. It allows us to react to new types of bots and new bot attacks. So it's critical that it is rolled out frequently and rapidly as bad actors change their tactics quickly.

A change in our underlying ClickHouse query behaviour (explained below) that generates this file caused it to have a large number of duplicate “feature” rows. This changed the size of the previously fixed-size feature configuration file, causing the bots module to trigger an error.

As a result, HTTP 5xx error codes were returned by the core proxy system that handles traffic processing for our customers, for any traffic that depended on the bots module. This also affected Workers KV and Access, which rely on the core proxy.

Unrelated to this incident, we were and are currently migrating our customer traffic to a new version of our proxy service, internally known as [FL2](#). Both versions were affected by the issue, although the impact observed was different.

Customers deployed on the new FL2 proxy engine, observed HTTP 5xx errors. Customers on our old proxy engine, known as FL, did not see errors, but bot scores were not generated correctly, resulting in all traffic receiving a bot score of zero. Customers that had rules deployed to block bots would have seen large numbers of false positives. Customers who were not using our bot score in their rules did not see any impact.

Throwing us off and making us believe this might have been an attack was another apparent symptom we observed: Cloudflare’s status page went down. The status page is hosted completely off Cloudflare’s infrastructure with no dependencies on Cloudflare. While it turned out to be a coincidence, it led some of the team diagnosing the issue to believe that an attacker may be targeting both our systems as well as our status page. Visitors to the status page at that time were greeted by an error message:

In the internal incident chat room, we were concerned that this might be the continuation of the recent spate of high volume [Aisuru DDoS attacks](#):

## The query behaviour change [↗](#)

I mentioned above that a change in the underlying query behaviour resulted in the feature file containing a large number of duplicate rows. The database system in question uses ClickHouse's software.

For context, it's helpful to know how ClickHouse distributed queries work. A ClickHouse cluster consists of many shards. To query data from all shards, we have so-called distributed tables (powered by the table engine `Distributed`) in a database called `default`. The `Distributed` engine queries underlying tables in a database `r0`. The underlying tables are where data is stored on each shard of a ClickHouse cluster.

Queries to the distributed tables run through a shared system account. As part of efforts to improve our distributed queries security and reliability, there's work being done to make them run under the initial user accounts instead.

Before today, ClickHouse users would only see the tables in the `default` database when querying table metadata from ClickHouse system tables such as `system.tables` or `system.columns`.

Since users already have implicit access to underlying tables in `r0`, we made a change at 11:05 to make this access explicit, so that users can see the metadata of these tables as well. By making sure that all distributed subqueries can run under the initial user, query limits and access grants can be evaluated in a more fine-grained manner, avoiding one bad subquery from a user affecting others.



The change explained above resulted in all users accessing accurate metadata about tables they have access to. Unfortunately, there were assumptions made in the past, that the list of columns returned by a query like this would only include the “default” database:

```
SELECT
    name,
    type
FROM system.columns
WHERE
    table = 'http_requests_features'
order by name;
```

Note how the query does not filter for the database name. With us gradually rolling out the explicit grants to users of a given ClickHouse cluster, after the change at 11:05 the query above started returning “duplicates” of columns because those were for underlying tables stored in the r0 database.

This, unfortunately, was the type of query that was performed by the Bot Management feature file generation logic to construct each input “feature” for the file mentioned at the beginning of this section.

The query above would return a table of columns like the one displayed (simplified example):

However, as part of the additional permissions that were granted to the user, the response now contained all the metadata of the r0 schema effectively more than doubling the rows in the response ultimately affecting the number of rows (i.e. features) in the final file output.

## Memory preallocation[🔗](#)

Each module running on our proxy service has a number of limits in place to avoid unbounded memory consumption and to preallocate memory as a performance optimization. In this specific instance, the Bot Management system has a limit on the number of machine learning features that can be used at runtime. Currently that limit is set to 200, well above our current use of ~60 features. Again, the limit exists because for performance reasons we preallocate memory for the features.

When the bad file with more than 200 features was propagated to our servers, this limit was hit — resulting in the system panicking. The FL2 Rust code that makes the check and was the source of the unhandled error is shown below:

This resulted in the following panic which in turn resulted in a 5xx error:

```
thread fl2_worker_thread panicked: called Result::unwrap() on an Err value
```

## Other impact during the incident[🔗](#)

Other systems that rely on our core proxy were impacted during the incident. This included Workers KV and Cloudflare Access. The team was able to reduce the impact to these systems at 13:04, when a patch was made to Workers KV to bypass the core proxy. Subsequently, all downstream systems that rely on Workers KV (such as Access itself) observed a reduced error rate.

The Cloudflare Dashboard was also impacted due to both Workers KV being used internally and Cloudflare Turnstile being deployed as part of our login flow.

Turnstile was impacted by this outage, resulting in customers who did not have an active dashboard session being unable to log in. This showed up as reduced availability during two time periods: from 11:30 to 13:10, and between 14:40 and 15:30, as seen in the graph below.

The first period, from 11:30 to 13:10, was due to the impact to Workers KV, which some control plane and dashboard functions rely upon. This was restored at 13:10, when Workers KV bypassed the core proxy system. The second period of impact to the dashboard occurred after restoring the feature configuration data. A backlog of login attempts began to overwhelm the dashboard. This backlog, in combination with retry attempts, resulted in elevated latency, reducing dashboard availability. Scaling control plane concurrency restored availability at approximately 15:30.

## Remediation and follow-up steps [🔗](#)

Now that our systems are back online and functioning normally, work has already begun on how we will harden them against failures like this in the future. In particular we are:

- Hardening ingestion of Cloudflare-generated configuration files in the same way we would for user-generated input
- Enabling more global kill switches for features

- Eliminating the ability for core dumps or other error reports to overwhelm system resources
- Reviewing failure modes for error conditions across all core proxy modules

Today was Cloudflare's worst outage [since 2019](#). We've had outages that have made our [dashboard unavailable](#). Some that have caused [newer features](#) to not be available for a period of time. But in the last 6+ years we've not had another outage that has caused the majority of core traffic to stop flowing through our network.

An outage like today is unacceptable. We've architected our systems to be highly resilient to failure to ensure traffic will always continue to flow. When we've had outages in the past it's always led to us building new, more resilient systems.

On behalf of the entire team at Cloudflare, I would like to apologize for the pain we caused the Internet today.

Time (UTC)	Status	Description
11:05	Normal.	Database access control change deployed.
11:28	Impact starts.	Deployment reaches customer environments, first errors observed on customer HTTP traffic.
11:32-13:05	The team investigated elevated traffic levels and errors to Workers KV service.	<p>The initial symptom appeared to be degraded Workers KV response rate causing downstream impact on other Cloudflare services.</p> <p>Mitigations such as traffic manipulation and account limiting were attempted to bring the Workers KV service back to normal operating levels.</p> <p>The first automated test detected the issue at 11:31 and manual investigation started at 11:32. The incident call was created at 11:35.</p>
13:05	Workers KV and Cloudflare Access bypass	During investigation, we used internal system bypasses for Workers KV and Cloudflare Access so they fell back to

	implemented — impact reduced.	a prior version of our core proxy. Although the issue was also present in prior versions of our proxy, the impact was smaller as described below.
13:37	Work focused on rollback of the Bot Management configuration file to a last-known-good version.	We were confident that the Bot Management configuration file was the trigger for the incident. Teams worked on ways to repair the service in multiple workstreams, with the fastest workstream a restore of a previous version of the file.
14:24	Stopped creation and propagation of new Bot Management configuration files.	We identified that the Bot Management module was the source of the 500 errors and that this was caused by a bad configuration file. We stopped automatic deployment of new Bot Management configuration files.
14:24	Test of new file complete.	We observed successful recovery using the old version of the configuration file and then focused on accelerating the fix globally.
14:30	Main impact resolved. Downstream impacted services started observing reduced errors.	A correct Bot Management configuration file was deployed globally and most services started operating correctly.
17:06	All services resolved. Impact ends.	All downstream services restarted and all operations fully restored.

Cloudflare's connectivity cloud protects [entire corporate networks](#), helps customers build [Internet-scale applications efficiently](#), accelerates any [website or Internet application](#), [wards off DDoS attacks](#), keeps [hackers at bay](#), and can help you on [your journey to Zero Trust](#).

Visit [1.1.1.1](#) from any device to get started with our free app that makes your Internet faster and safer.

To learn more about our mission to help build a better Internet, [start here](#). If you're looking for a new career direction, check out [our open positions](#).

[Outage](#) [Post Mortem](#) [Bot Management](#)

---

## Follow on X

Matthew Prince | [@eastdakota](#)

Cloudflare | [@cloudflare](#)

---

## RELATED POSTS

October 28, 2025 12:00 PM

### Online outages: Q3 2025 Internet disruption summary

In Q3 2025, we observed Internet disruptions around the world resulting from government directed shutdowns, power outages, cable cuts, a cyberattack, an earthquake, a fire, and technical problems, as well as several with unexplained causes....

**By** David Belson

[Radar](#), [Internet Shutdown](#), [Internet Traffic](#), [Outage](#), [Consumer Services](#), [Internet Trends](#)

September 30, 2025 10:05 AM

### Nationwide Internet shutdown in Afghanistan extends localized disruptions

On September 29, 2025, Internet connectivity was completely shut down across Afghanistan, impacting business, education, finance, and government services....

**By** David Beison

[Radar](#), [Internet Shutdown](#), [Internet Traffic](#), [Outage](#)

September 24, 2025 1:10 PM

## Giving users choice with Cloudflare's new Content Signals Policy

Cloudflare's Content Signals Policy gives creators a new tool to control use of their content. ...

**By** Will Allen

[Birthday Week](#), [Policy & Legal](#), [Bot Management](#), [AI](#)

September 23, 2025 2:00 PM

## Building unique, per-customer defenses against advanced bot threats in the AI era

Today, we are announcing a new approach to catching bots: using models to provide behavioral anomaly detection unique to each bot management customer and stop sophisticated bot attacks. ...

**By** Jin-Hee Lee, Oliver Payne, Bob AminAzad, Viktor Chynarov, Aleksandar Pavlov Hrusanov, Prajjwal Gupta

[Birthday Week](#), [AI](#), [Bots](#), [Bot Management](#)

