CLOUDFLARE®

Enterprise Best Practices

Optimizing Your Performance and Security Posture

1 888 99 flare | success@cloudflare.com | www.cloudflare.com

Table of Contents

PERFORMANCE

CDN

Web Optimization

DNS

Load Balancing

Argo

China Network

SECURITY

DDoS

SSL

WAF

Rate Limiting

Argo Tunnel

Table of Contents

INSIGHTS & ADMINISTRATION

Threat Analytics

Enterprise logs

// Performance
Best Practices

RECOMMENDATIONS

- □ Accelerate content delivery
- □ Cache as much static and semi-static content as possible
- □ Purge the cache via API for event-driven content
- □ Accelerate dynamic content with Railgun
- □ Enable Argo for tiered caching and smart routing
- □ Reduce latency, boost availability with Load Balancing
- □ Ensure optimal performance on our China Network

Cloudflare can accelerate all of the content you are trying to deliver to your users. By optimizing your images and being as near as possible to your end users, Cloudflare's performance suite provides the fastest experience for your application.

1. Accelerate content delivery

Speed

The features within our Speed application work by delivering your content more intelligently. Some of these features are client-side (implemented in the browser using javascript), while others are edge-side (performed at Cloudflare's edge before content leaves a Cloudflare data center).

□ Enable Polish for image compression

Polish offers compression rates up to 30-40% with either lossy or lossless compression. Polish should always be enabled in the lossless, 'Basic' mode, and even lossy for the majority of web applications.

□ Enable Minification for HTML and CSS

Auto-Minify removes comments and formatting meant for humans which leads to improved load times. Minification can be safely enabled for HTML and CSS. For Javascript, please make certain that line endings are denoted with a semicolon.

□ Test Mirage and Rocket Loader with your system

Cloudflare can even accelerate your content using Client-side tools such as Mirage and Rocket Loader. These features are currently in Beta, and we recommend testing them carefully before turning them on for production traffic. To test Mirage or Rocket Loader, you can enable them for specific pages or subdomains with page rules.

□ Test Mirage and Rocket Loader with your system (cont.)

- Mirage is a mobile specific image optimization solution that “right-sizes” images based on the screen resolution, lazy loads images below the fold, bundles images and asynchronously loads them after page load to reduce the time until a mobile user can interact with a page.

- Rocket Loader grabs all javascript and forces it to be loaded asynchronously after the page load. This feature is powerful and aggressive, ideal for customers who can’t manage the loading order of the javascript on a page.

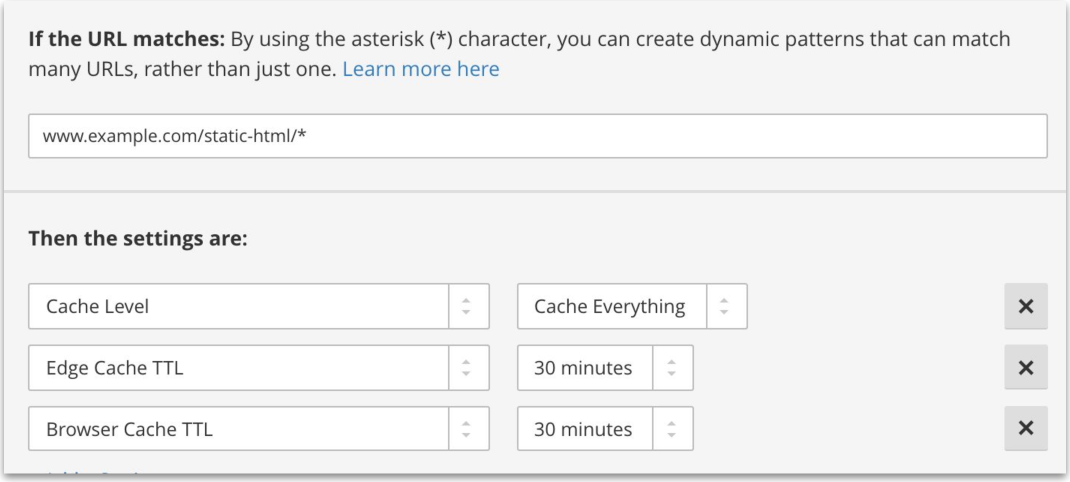
2. Cache as much static and semi-static content as possible

Page Rules

Your goal should be to cache as much static (and semi-static) content as possible in order to leverage our globally distributed network to its fullest. By default, Cloudflare will only cache content we can be certain is static in nature, such as images, javascript, and CSS files. Our default, and recommended, caching mode includes query parameters in the request.

□ Enable Cache Everything for static HTML webpages

This can be turned on for each path in the Page Rule application:



If the URL matches: By using the asterisk (*) character, you can create dynamic patterns that can match many URLs, rather than just one. Learn more here

www.example.com/static-html/*

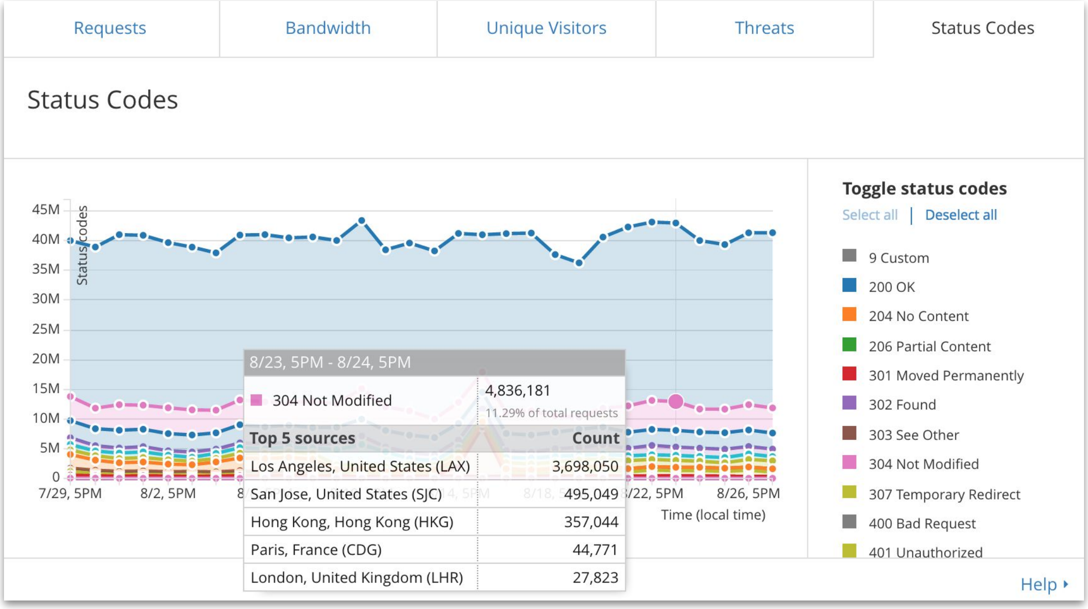
Then the settings are:

Cache Level	Cache Everything	✕
Edge Cache TTL	30 minutes	✕
Browser Cache TTL	30 minutes	✕

1. Create a Page Rule with the URL pattern of your static HTML (i.e. www.example.com/static-html/*)
2. Set Cache Level to ‘Cache Everything’
3. Set Edge and Browser TTLs
4. Press ‘Save and Deploy’

❑ Utilize conservative TTLs (Times-to-Live) for content that rarely changes

For content that changes only occasionally, you can set a conservative TTL to utilize our cache as much as possible. A good way to tell if your TTLs may need to be adjusted is by watching your Status Codes in our Analytics App for an abundance of 304 requests. If you have a high percentage of re-validation requests, you could likely increase the TTLs of your content without negatively impacting your customers. This will use our cache more effectively and increase performance since you will revalidate less often.



8/23, 5PM - 8/24, 5PM

304 Not Modified — 4,836,181 — 11.29% of total requests

Top 5 sources	Count
Los Angeles, United States (LAX)	3,698,050
San Jose, United States (SJC)	495,049
Hong Kong, Hong Kong (HKG)	357,044
Paris, France (CDG)	44,771
London, United Kingdom (LHR)	27,823

How do I tell if items are being cached?

Cloudflare adds the response header “CF-Cache-Status” if attempting to cache the object. The value of this header indicates if successful:

- MISS: Not yet in the cache or the TTL expired (i.e. cache-control max age of 0)
- HIT: Asset delivered from cache
- EXPIRED: Resource was in cache but has since expired, served from origin server
- REVALIDATED: Delivered from cache. The TTL was expired, but a “If-Modified-Since” request to the origin indicated the asset has not changed so the version in cache is considered valid again.

3. Purge the cache via API for event-driven content

Caching

For example, every time a new post is added to your blog, you could easily purge the Cloudflare cache using an API command. It is very common to see event-driven content, and we make it easy to ensure no stale content is reaching your users. The necessary commands are listed below to purge the cache immediately across our entire global network either via our Caching Application or via the API.

Additional recommendations for cache purging...

1 Purge the cache for individual files

Purging individuals objects is a great way to maintain your cache-hit ratio while still ensuring certain objects are re-validated in our cache. API Documentation

2 Purge the cache by Cache-Tag

Cache-Tags allow you to define buckets of content that you wish to purge all together. This is an excellent way to combine objects that are commonly changed together. So an HTML blog post, for example, and all of its image content could be tagged together. Mobile-only content could also be bundled using cache-tags to purge everything when you push a new update to your mobile domain.

3 Purge the cache globally

You also have the option to force our entire cache to revalidate. This allows you to reset all of the objects stored in our cache to ensure every request will return to the origin.

4 Purge the cache by Page Rule

Page Rules

Page Rules allow you to effectively purge the entire cache by a basic regular expression. These let you utilize a pre-defined Page Rule and re-validating all hits against that Page Rule.

Advanced Enterprise Caching Features

❏ Bypass Cache on Cookie

The ability, configured in a page rule, to serve a cached object unless we see a cookie of a specific name, e.g., serve a cached version of the homepage unless we see a SessionID cookie indicating the customer is logged in and therefore should be presented personalized content.

❏ Cache on Cookie

Presents a cached page when only when we see a specific cookie, e.g., only serves a cached page once a device type cookie has been set by the origin server.

❏ Custom Cache Keys

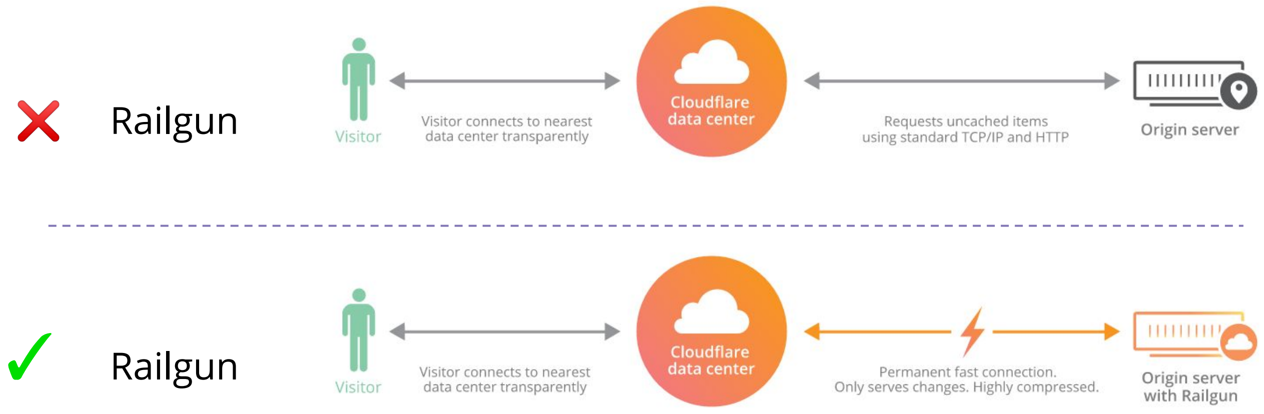
Generally, objects in Cloudflare's cache are referenced by only their URI (e.g. https://www.example.com/logo.png). We offer the ability to create custom cache keys so that a different object is served for the same URI based on any arbitrary request header or cookie. For example, https://www.example.com/logo.png with a device type cookie set to *desktop* would be a different object in our cache than https://www.example.com/logo.png with a device type cookie set to *tablet*.

4. Accelerate dynamic content with Railgun

Speed

Web applications often render personalized, dynamic content which means that this content must be proxied through to the origin server. Railgun is an optional feature available to Business & Enterprise accounts that accelerates requests for this type of personalized content.Railgun is fundamentally a de-duplicating proxy.

The railgun listener is installed within the customer's origin infrastructure so that before a response is sent from the origin infrastructure, a binary diff is performed with the last resource at the same URI. For example, when Bob attempts to access his shopping cart page, the application server renders it, but before it leaves the origin datacenter, the railgun listener compares Bob's page with Alice's page, and only sends the difference. Documentation


✗ Railgun
Visitor
Visitor connects to nearest data center transparently
Cloudflare data center
Requests uncached items using standard TCP/IP and HTTP
Origin server
✓ Railgun
Visitor
Visitor connects to nearest data center transparently
Cloudflare data center
Permanent fast connection. Only serves changes. Highly compressed.
Origin server with Railgun

Railgun Best Practices

- ❑ Consider high-availability (HA) configuration

  As Cloudflare will fall back to HTTP in the event railgun fails, a high availability solution is generally not required. If high availability is a preferred solution, all railgun instances should be placed behind a layer 3 load-balancer and registered with the same activation code as seen in the diagram. If configured in an active/active configuration, Railguns should share memcache to improved cache hit rate. If configured in an active/passive configuration, Railguns can have independent memcaches.

- ❑ Railgun should be outside of a load balancer if possible

  The railgun listener (stand-alone or cluster) should logically sit outside of the load balancer allowing requests for all application servers to flow through the same railgun instance increasing the cache hit ratio.

- ❑ Keep Railgun near the origin

  The railgun listener should be less than 5 milliseconds from your application server.

- ❑ Railgun can easily be created within AWS

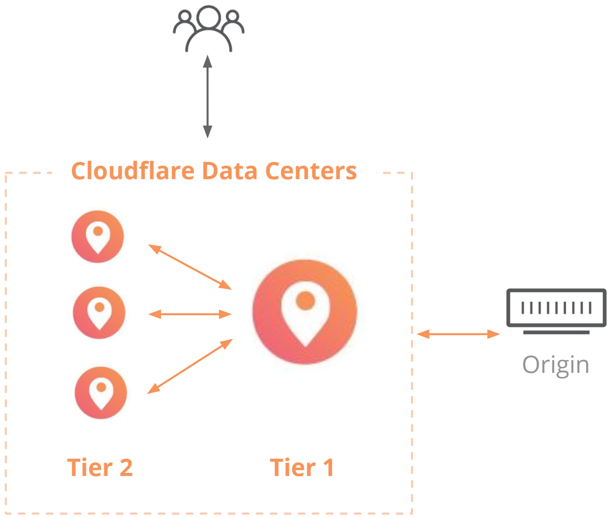
  We've provided a link to some excellent documentation from our friends at AWS.

5. Enable Argo for tiered caching and smart routing

Traffic

Our Argo product suite is comprised of three distinct features: Tiered Caching, Smart Routing, and Tunnel. While all three features are beneficial for performance, Tunnel is primarily a Security product, so you can find best practices in that section of the guide. Tiered Caching and Smart Routing, once enabled, “just work” -- no special configuration is needed. More information below:

□ Tiered Caching is enabled by default for all enterprise domains

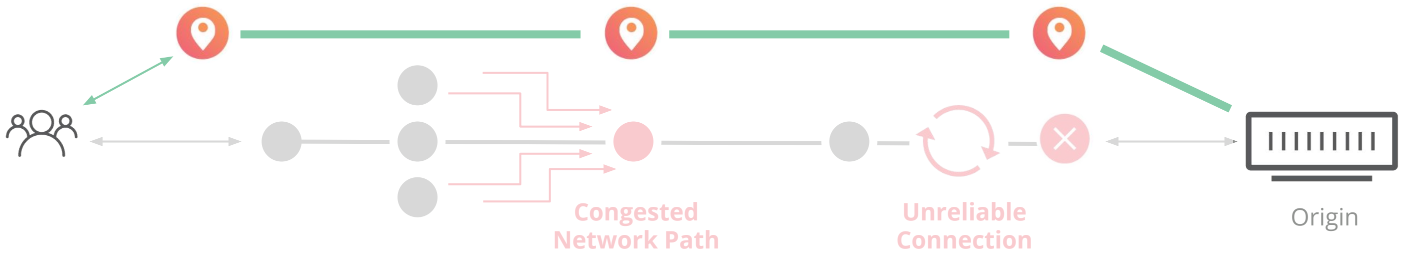


Argo Tiered Caching uses the size of our network to reduce requests to your origin by dramatically increasing cache hit ratios. It is on by default for all enterprise zones and geographically-crafted to allow for optimal performance.

How it works: When a cache miss occurs as one of our data centers, instead of going directly back to your origin to retrieve the requested content, we will first ask other Cloudflare PoPs if they have the content in cache. This results in improved performance for visitors and reduced load on your origin.

□ Ensure Smart Routing is enabled for dynamic content acceleration

Smart Routing analyzes and optimizes routing decisions across the global Internet in real time, steering your traffic around congested or unreliable network paths.



You can turn Smart Routing on in the Traffic app. It is a non-intrusive change that will not cause downtime, but you may need to let it run for up to a few hours in order to find all of the optimal paths to your origin across the globe.

6. Reduce latency, boost availability with Load Balancing

Traffic

Over-utilized or geographically distant servers add unnecessary latency and degrade the visitor experience. Cloudflare's load balancing solution allows you to route traffic to the closest geolocation regions to your users, and to regularly check the health of your origins to route visitors away from failures.

Our Knowledge Base contains detailed setup instructions, which you can supplement with the following best practices:

□ Begin by creating a load balancer serving non-production traffic

As with many features, we recommend thorough testing before enabling load balancing on production traffic. That means creating a load balancing configuration with test traffic to understand the effects on your origin servers before going live in production.

Tips for Properly Configuring Health Checks

1. Match the encryption of your production traffic

   If your production traffic is served over HTTP, configure your health checks over HTTP. If HTTPS, send health checks over HTTPS. This will ensure your health checks provide an accurate representation of live traffic hitting your origin servers.

2. Set a health check interval appropriate for your origins

   When deciding on the intervals at which Cloudflare's load balancer should send health checks to your origin, consider how quickly your origins are capable of handling these requests.

3. Send health checks from appropriate regions

   Cloudflare's network is very large, and growing. Before you configure health checks to be sent from "All Data Centers", ensure your origins are prepared to be checked by every Cloudflare network node at the interval you configure.

   Alternatively, you can set up health checks from specific regions, which will limit these checks to a subset of our data centers.

□ Avoid duplicate configurations

If you have a load balancing configuration that you want to share with additional domains, there is no need to recreate the load balancer for each one. Instead, you can simply create a CNAME record in your Cloudflare DNS app to do so.

Another popular configuration option is sharing the same monitors and pools between domains, but configuring separate load balancers. This is beneficial for use cases in which you want to vary the failover priority among your domains.

6. Ensure optimal performance on our China Network

Network

Through our partnership with Baidu, Cloudflare customers can serve content to their visitors within China from a network of more than twenty data centers across mainland China. Enabling our China network will significantly reduce latency within the country, but due to China's unique internet architecture, the following best practices are essential for optimizing performance.

□ Lighten your content load

One way to approach optimizing performance for China is to consider the strategy you might use for mobile users in the rest of the world. The network difficulties are similar. The key is to focus on serving the lightest possible load. In particular, consider relieving your site of all non-essential third-party resources.

Extensive testing is critical to successful performance inside China. Testing will help you uncover unnecessary third-party resources being served, determine how to improve your cache hit rate, and lead to many other ideas for reducing latency

Fast Google Fonts with Cloudflare Workers

Google Fonts - one of the most common third-party resources in use - is a perfect example of a resource that can really slow your site down in China.

Check out this detailed tutorial on our blog that shows how you can use Cloudflare Workers to make your Google Fonts lightning-fast.

❑ Cache as much content as possible

Beyond serving up a “light” site, the key to fast performance on our China Network is boosting your cache hit rate as high as possible. Cloudflare caches many file extensions by default, but you will want to create Cache Everything page rules to ensure static HTML and all other static assets can be served from our in-country caches.

Set your Edge TTLs in Cloudflare as high as you are willing to go, and set your browser TTLs for as long as you expect content to stay fresh.

20+ points of presence across mainland China

Analyze your logs (and work with your dedicated account team) to uncover the top un-cached files being served. Can any of these be cached?

❑ Enable all applicable Cloudflare speed features

✓ **Brotli** will greatly reduce the write time transfer. If multiple compression methods are supported by the client, Cloudflare will select Brotli compression as the preferred content encoding method. If the client does not indicate that Brotli compression is supported, then gzip compression will be applied.

✓ **Polish** reduces image file size by removing metadata and compressing images when possible. We recommend enabling WebP, and unless image quality is central to your business, turn on Polish in Lossy mode for China.

✓ **Rocket Loader** will also be helpful, as it improves paint times by asynchronously loading your Javascripts, including third party scripts, so that they do not block rendering the content of your pages.

RECOMMENDATIONS

- ❑ Secure origin IP addresses
- ❑ Configure your Security Level selectively
- ❑ Activate your Web Application Firewall safely
- ❑ Leverage Rate Limiting for granular layer 7 protection
- ❑ Build Firewall Rules for fine-grained control
- ❑ Use SSL for SaaS to extend Cloudflare to your customers
- ❑ Easily configure a secure origin connection with Argo Tunnel

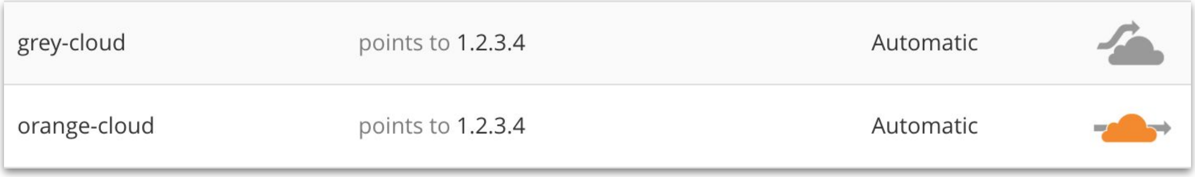
By default, Cloudflare's security settings are set to safe defaults that aim to avoid false positives and negative influences on your traffic. However, this is not necessarily the best security posture for every Enterprise customer. The following steps will ensure Cloudflare is configured in a secure and safe manner.

1. Secure origin IP addresses

❑ Orange-cloud all DNS records for HTTP(S) traffic from your origin

DNS

When a subdomain is orange-clouded within our DNS application, Cloudflare will actively proxy that traffic by responding with Cloudflare IP addresses. These addresses cause the client to connect to Cloudflare first and obscure the origin IP address. To improve the security of your origin IP address, all HTTP(s) traffic should be orange-clouded.



grey-cloud	points to 1.2.3.4	Automatic	
orange-cloud	points to 1.2.3.4	Automatic	

❑ Obscure grey-clouded origin records with non-standard names

Any records that cannot be proxied through Cloudflare but still utilize your origin IP such as FTP — can still be secured with additional obfuscation. If you require a record to your origin that cannot be proxied by Cloudflare, use a non-standard name for this record. For example, instead of ftp.example.com use [random word or-random characters].example.com — this will make dictionary scans of your DNS less likely to expose your origin IP addresses.

□ Separate IP ranges for HTTP and non-HTTP traffic if possible

Some customers will use separate IP ranges for HTTP and non-HTTP traffic, allowing them to orange-cloud all records pointing to their HTTP IP range and obscuring all non-HTTP traffic with a different IP subnet.

2. Configure your Security Level selectively

Firewall

Cloudflare sees nearly 2 billion unique IPs every month from more than 13 million websites on our network. Each IP is assigned a threat score, and as an IP engages in malicious behavior on our network, its threat score increases. The scale of this data, and the automatic process of assigning threat scores, allows Cloudflare to quickly find bad actors and prevent them from reaching your assets.

□ Increase for sensitive admin and login pages

Page Rules

Customers often create Page Rules to heighten the Security Level on admin and login pages in order to prevent brute-force attempts.

1. Ensure the proper domain is selected
2. Create a Page Rule with the URL pattern of your API (i.e. www.example.com/wp-login)
3. Select the ‘Security Level’ setting
4. Mark the setting as ‘High’
5. Select ‘Save and Deploy’

□ Decrease for non-sensitive paths or APIs

Page Rules

To reduce the risk of false positives, this setting can be decreased for general pages and API traffic.

1. Ensure the proper domain is selected
2. Create a Page Rule with the URL pattern of your API (i.e. www.example.com/api/*)
3. Select the ‘Security Level’ setting
4. Mark the setting as ‘Low’, ‘Off’, or ‘Essentially Off’
5. Press ‘Save and Deploy’

Security Level Options

Each Security Level is aligned with a threat score from our IP Reputation Database. A threat score above 10 is considered risky. A threat score above 50 indicates real hazard.

HIGH - Threat scores greater than 0 will be challenged.

MEDIUM - Threat scores greater than 14 will be challenged.

LOW - Threat scores greater than 24 will be challenged.

ESSENTIALLY OFF - Threat scores greater than 49 will be challenged.

OFF - Enterprise customers can turn off this feature entirely.

□ Alternatively, utilize the 'Medium' Security Level setting globally

Firewall

Cloudflare's 'Medium' Security Level has a very low false positive rate (1 in 50 million) and is recommend as a starting point for all customers. The global setting is available in the Firewall Application.

1. Ensure the proper domain is selected
2. Select the FireWall application
3. Within the Security Level module, click 'Options'
4. Select 'Medium'

3. Activate your Web Application Firewall safely

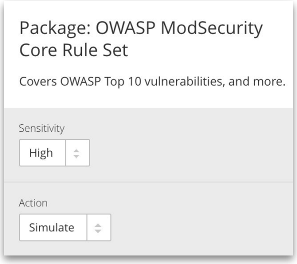
Firewall

Your WAF is available in the Firewall Application in the Web Application Firewell section. We will walk through these settings in reverse to ensure that the WAF is configured as safely as possible before turning it on for your entire domain. The goal of these initial settings is to reduce false positives and to populate the Traffic Application with WAF events for further tuning.

□ Set the OWASP ModSecurity sensitivity to High with an action of Simulate

The OWASP package is based on an anomaly score to help reduce false positives while still catching attacks. The goal of these settings is to tune your OWASP settings by logging any false positives.

A Simulate action simply logs an event within our Traffic app.



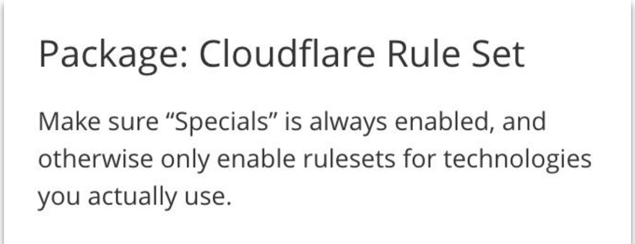
Package: OWASP ModSecurity Core Rule Set

Covers OWASP Top 10 vulnerabilities, and more.

Sensitivity: High

Action: Simulate

□ Activate Cloudflare Specials and any platform-specific groups you use



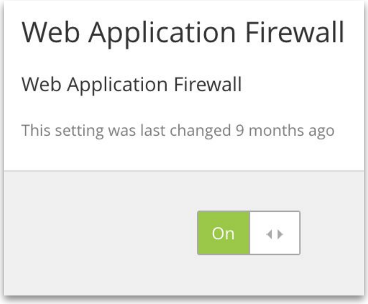
Package: Cloudflare Rule Set

Make sure "Specials" is always enabled, and otherwise only enable rulesets for technologies you actually use.

The Cloudflare Rulesets are focused on novel, zero-day, and application-specific exploits. Selecting Cloudflare Specials and any application specific groups allow you to take an adequate security posture with very few false positives.

For example, if you use a Drupal application, that group should be activated while Joomla and Magento groups are turned Off.

□ Finally, turn your WAF On with the Global Setting



Web Application Firewall

Web Application Firewall

This setting was last changed 9 months ago

Now that your Package-level settings are configured safely, you can now turn on the global, domain-wide WAF.

If you are concerned about interfering with legitimate requests (e.g. to an API endpoint), you can use Page Rules to disable the WAF for specific paths. This can ensure you are protected while carving out areas of your zone where the WAF should not be activated.

4. Leverage Rate Limiting for granular layer 7 protection

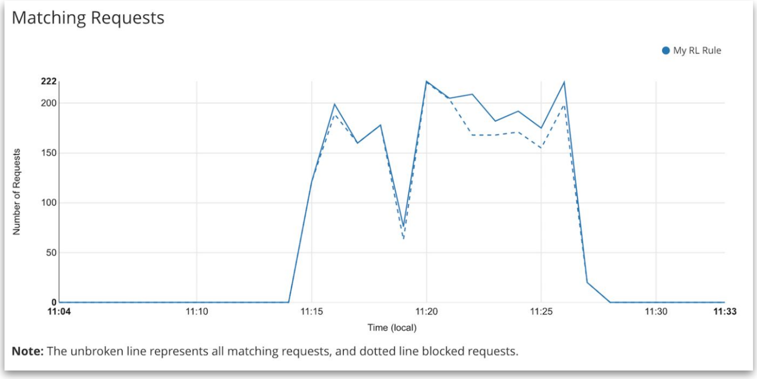
Firewall

Rate Limiting complements Cloudflare's advanced DDoS protection by allowing for precise mitigation of the most sophisticated attacks against the application layer. Most commonly, customers implement rate limits to prevent brute-force login attacks, to limit access to high-cost resources, and for greater precision in DDoS attack prevention.

Our Knowledge Base includes detailed instructions for creating various rate limiting rules. Below are some best practices we developed with our Enterprise customers:

□ Use Cloudflare analytics to define "normal" traffic patterns

To avoid interfering with good traffic to your site or application, you will want to set your rate limiting thresholds comfortably (at least 4x) above typical request levels.



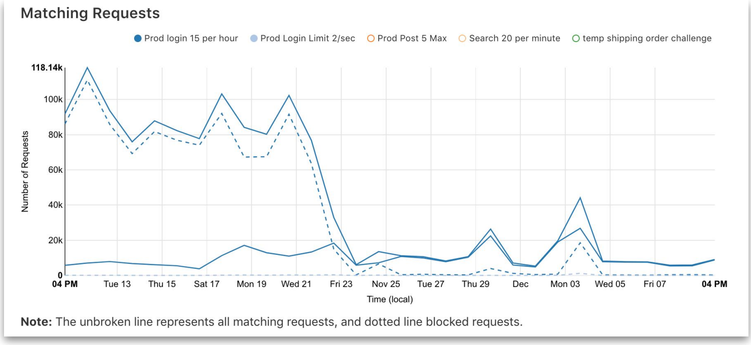
Matching Requests

Note: The unbroken line represents all matching requests, and dotted line blocked requests.

If you are unsure what normal levels are, configure rate limits in Simulate Mode.

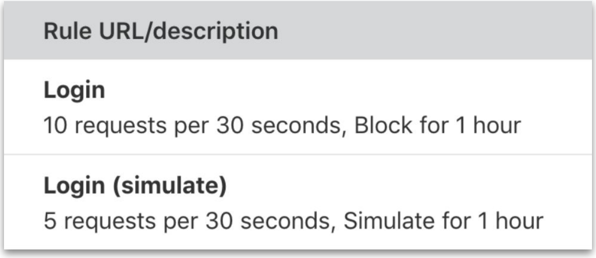
The simulation will show you how many requests would have been blocked if you had enabled the rules. You can then adjust your limits accordingly.

❑ Stack rules for security from more advanced attacks



Note: The unbroken line represents all matching requests, and dotted line blocked requests.

Rate limiting rules execute simultaneously (unlike page rules). That allows you to create overlapping rules on the same path. Among many reasons this is useful, this complexity helps guard against sophisticated attackers that attempt to determine your rate limiting thresholds.



Rule URL/description
Login 10 requests per 30 seconds, Block for 1 hour
Login (simulate) 5 requests per 30 seconds, Simulate for 1 hour

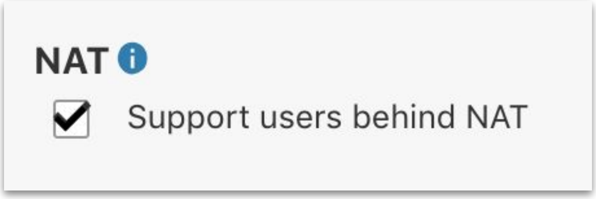
Combining simulated rules with active rules is another useful form of stacking: setting the active rule at a safe threshold and the simulated rules at more aggressive thresholds.

If you come under attack and need to make your rules more aggressive, you can review your simulated analytics to determine how these rule changes will affect your live traffic before enabling them.

❑ Consider support for users behind a NAT

Cloudflare rate limits requests per IP address. That could be a problem if you expect traffic from many users that share a single IP address, such as those on corporate networks or Carrier-Grade NATs.

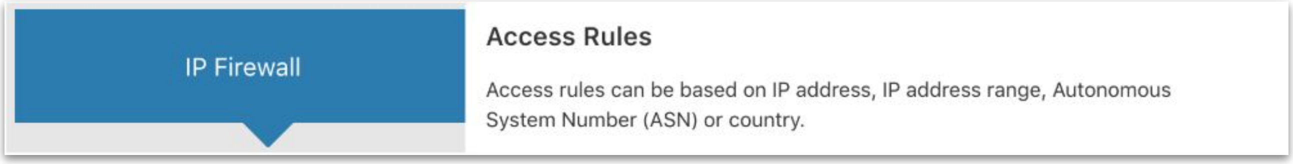
To solve for this use case, you can enable the “Support users behind NAT” option, which can be set per rate limiting rule. This will allow Cloudflare to set a cookie that identifies unique users behind the NAT.



NAT ⓘ

☑ Support users behind NAT

❑ Review logs and use the IP Firewall to block repeat offenders



IP Firewall

Access Rules

Access rules can be based on IP address, IP address range, Autonomous System Number (ASN) or country.

Use your logs to learn from rate limited events and improve your security posture. If you encounter an IP address appearing repeatedly, consider adding it to your Cloudflare IP Firewall.

6. Build Firewall Rules for fine-grained control

Firewall

The comprehensive managed rules within our WAF are automatically updated by our engineers to keep you protected from new, advanced threats. You can also use our IP Firewall to manually block or challenge IPs, countries, and ASNs that you know to be malicious. Firewall Rules give you greater power and flexibility to examine and take action on your traffic. You can define a filter using multiple, custom criteria and declare a specific action to apply when that filter is matched.

Access our Developer Portal for detailed documentation on configuring Firewall Rules.

❑ Create custom WAF rules at the edge for specific paths

Cloudflare’s WAF is applied globally. Once enabled, the rules are triggered on all of your orange-clouded records. If any part of your site is incompatible with specific WAF rules, you can instead use Firewall Rules to apply a rule only to specific paths.

❑ Easily implement “positive” security controls

The above is an example of a “negative” security control, in which you allow all traffic by default, but create a rule to block or challenge a specific type of traffic. But Firewall Rules enable you to implement a “positive” security control, in which all traffic is blocked by default, but you allow access to specific types of traffic, in line with the trend towards a “Zero Trust” security model.

To take just one example, you can create a Firewall Rule to block all traffic to a specific subdomain on your website *except for* traffic from a specific IP range - i.e. from your employees. Or you can challenge all requests to a login API endpoint *except for* those made via API.

4. Add custom hostnames to Cloudflare with SSL for SaaS

Cloudflare's SSL for SaaS offering allows you to extend all of Cloudflare's performance and security features to your customers' vanity domains (or any custom hostname), including DDoS protection, CDN and content optimization, and SSL certificate provisioning and renewal. All your customers need to do is add the initial CNAME to your domain. From there, you send a single API call and Cloudflare takes care of the rest.

Below are a few best practices we recommend, but contact your dedicated account team to access the SSL for SaaS Onboarding Guide for detailed instructions.

□ Reduce downtime risk by pre-provisioning certificates

If you are migrating any customers' vanity domains that already have HTTPS (perhaps they're self-hosted or you've manually provisioned), these customers will likely want to avoid the few minutes of downtime during the cutover process.

To minimize the risk of downtime, you can employ additional "pre-validation" methods to obtain a certificate in advance of setting the CNAME to your domain:

Pre-Validation Methods

1. HTTP-based validation *(Recommended)*

- HTTP-based validation - manually hosting the DCV token at your origin server - is the most common validation method, in part because it does not require involvement from your end customer.
- Reference the SSL for SaaS Onboarding Guide (sent to you by your account team) for complete instructions.

2. DNS CNAME-based validation

- Specify "method":"cname" in your API call, and the response will contain a CNAME to set up in your authoritative DNS.
- Once you have set that CNAME and can see that the record has propagated, send a PATCH to have the certificate issued.

3. Email-based validation

- An email will be sent to the contacts listed for the domain in WHOIS.
- When the domain owner has opened the link in this email and clicked "Approve" on the validation page, the certificate will automatically transition through pending to activation.

□ Stage your rollout in phases for smooth provisioning

By default, Cloudflare can successfully order up to 15 certificates per minute. To ensure you don't exceed that rate limit, stage your SSL for SaaS rollout in phases. If you need this limit raised for a specific reason, please contact your dedicated account team.

□ Use custom origins to segment your customers

By default, Cloudflare will send all of your custom hostname traffic to the proxy fallback record you provide. But that's not your only option. If you have typically segmented your customers to send traffic to different origin servers (e.g. by region or tier), you can configure your SSL for SaaS setup to continue doing so.

To enable custom origins, please contact your dedicated account team, who can entitle this feature and walk you through the configuration process.

Note: Your custom origins must be added to the DNS panel in your Cloudflare account in order to be used.

8. Easily secure your origin connection with Argo Tunnel

Finally, we recommend implementing Argo Tunnel in order to secure the connections from your origin to Cloudflare. Argo Tunnel offers an easy way to expose web servers securely to the internet, without opening up firewall ports and configuring ACLs. As requests will then route through Cloudflare before reaching the web server, you can be sure attack traffic is stopped with Cloudflare's WAF and Unmetered DDoS mitigation (and authenticated with Access if you've enabled that feature).

As the name suggests, Argo Tunnel utilizes Cloudflare's Argo network, and is included in your Argo subscription. This means enabling Argo Tunnel will not only improve your origin security, but performance, as well.

For complete enablement instructions, please visit our Developer Portal.

Note: As a daemon installed at your origin providing the benefits of delta compression, Argo Tunnel can be thought of as a replacement for Cloudflare's Railgun technology. If you are enabling Argo Tunnel, you will not need to run Railgun in parallel.

RECOMMENDATIONS

- ❑ Ingest raw request logs for detailed analytics
- ❑ Visualize logs with your preferred analytics provider
- ❑ Enforce 2-Factor Authentication for your organization
- ❑ Manage your brand by customizing Cloudflare error pages

1. Ingest Cloudflare logs for detailed analytics

The Analytics app in the Cloudflare dashboard is designed to provide insights into your traffic at a high level. As an Enterprise customer, you also have the ability to ingest all of your HTTP request logs, either by pulling these logs via API or having Cloudflare push them to you:

❑ Logpull API

Logpull is a REST API for downloading request logs over HTTP. To enable it, contact your Customer Success Manager and ask for Logpull to be turned on (your logs are not retained by default).

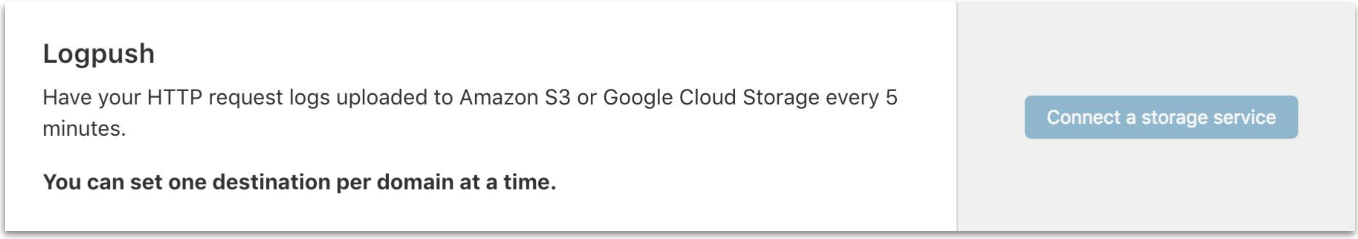
Our developer documentation provides detailed instructions for using the API.

❑ Logpush service

Your can also have your request logs pushed to you at a set interval (every 5 minutes) using the Cloudflare Logpush service. All Logpull functionality — including selecting fields and sampling — is also available in Logpush.

Currently, you can have your logs pushed to Amazon S3 and Google Cloud Storage. Cloudflare will be expanding the available destinations on an ongoing basis.

Contact your Customer Success Manager to enable Logpush. Then, navigate to Analytics > Logs and begin the configuration process:



Logpush

Have your HTTP request logs uploaded to Amazon S3 or Google Cloud Storage every 5 minutes.

You can set one destination per domain at a time.

2. Visualize logs with your preferred analytics provider

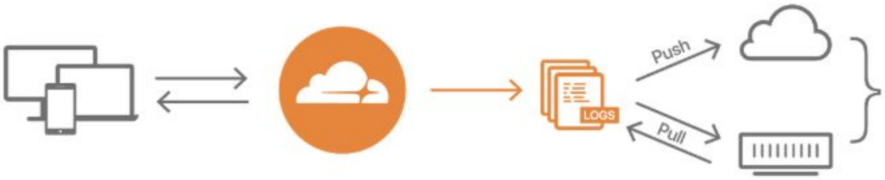
Once you've ingested your raw request logs from Cloudflare—either via Logpull or Logpush—we recommend you use your analytics provider of choice to visualize your data.

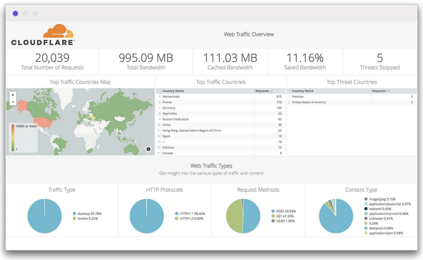
❑ Use Cloudflare partners for visualization, monitoring and alerting

We've made it easy to visualize logs by partnering with analytics providers such as Sumo Logic and Looker. A metrics-based integration with Datadog is also available, and more integrations are on the way. If your analytics provider is not yet on our list of partners, contact your Customer Success Manager to inquire about adding them to our list of partners.

For those providers with whom we partner, simply install the Cloudflare app—found in your analytics provider's integration catalogue or app store—to get access to the pre-built Cloudflare dashboards. Use these dashboards for advanced analytics, monitoring and alerting, or customize your own.

You can also combine Cloudflare logs and metrics with other data, such as your origin server logs, to get unique insights and better end-to-end visibility.





CLOUDFLARE

Web Traffic Overview

20,039 Total Number of Requests | 995.09 MB Total Bandwidth | 111.03 MB Cached Bandwidth | 11.16% Saved Bandwidth | 5 Threats Stopped

Top Traffic Countries

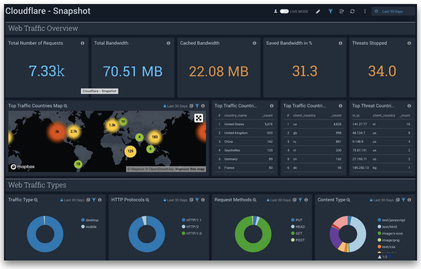
	Country Name	Requests
5	Netherlands	810
6	France	370
7	Germany	139
8	Seychelles	63
9	Russian Federation	62
10	China	48
11	Hong Kong, Special Admin Region of China	22
12	Spain	18
13	⌀	18
14	Pakistan	10
15	Canada	8

Top Threat Countries

	Country Name	Requests
1	Pakistan	3
2	United States of America	2

Web Traffic Types

Get insight into the various types of traffic and content



Cloudflare - Snapshot

Web Traffic Overview

Total Number of Requests 7.33k | Total Bandwidth 70.51 MB | Cached Bandwidth 22.08 MB | Saved Bandwidth in % 31.3 | Threats Stopped 34.0

Cloudflare - Snapshot
Top Traffic Countries Map
Last 30 Days
3k
2.7k
1.3k
10
4
183
129
6
18
mapbox
© Mapbox © OpenStreetMap Improve this map

Top Traffic Countri...

#	country_name	_count
1	United States	5,676
2	United Kingdom	929
3	China	162
4	Seychelles	129
5	Germany	89
6	France	80

Top Traffic Countri...

#	client_country	_count
1	us	4,825
2	gb	959
3	ru	861
4	nl	200
5	cn	162
6	de	95

Top Threat Countri...

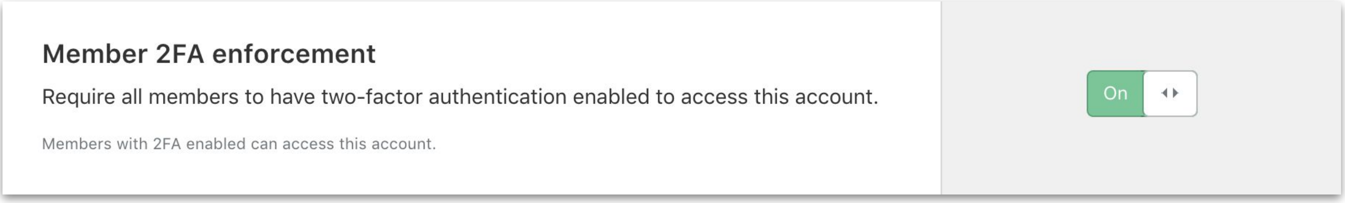
nt_ip	client_country	_count
141.27.77	nl	16
46.134.7	us	8
9.148.9	us	4
75.87.101	us	2
27.195.71	us	2
185.250.13	bg	1

Web Traffic Types

3. Enforce 2-Factor Authentication for your organization

Cloudflare allows you to enforce 2-Factor Authentication. This extra element of security can prevent unauthorized access to your Cloudflare console.

1. First, click on the name of your Account in the top left corner of the dashboard navigation and select Account Home
2. From your Account Home page, navigate to the Members tab, where you'll find the option to Enforce 2-Factor Authentication:



Member 2FA enforcement

Require all members to have two-factor authentication enabled to access this account.

Members with 2FA enabled can access this account.

4. Manage your brand by customizing Cloudflare pages

Custom P...

Cloudflare has to occasionally show content to your end-users. This can include security challenge pages, block pages, and error pages. All of this content can be customized by you to ensure a consistent branding experience for your customers. There are two ways to customize the content on these pages:

❏ Domain-specific custom pages

- After selecting the domain from the top-level navigation, select the Custom Pages app. From here, you can customize the pages shown to your visitors for several different security rules and error codes. These pages will only be shown to visitors of this domain.

❏ Account-wide custom pages

- To apply the same custom pages across your Cloudflare domains, simply navigate to your Account Home (by selecting the name of your account from the top left corner of the dashboard navigation).
- From there, select Configurations → Custom Pages. Any custom pages you configure here will apply across all domains in that account.
- *Note: If you have setup more than one account in Cloudflare to organize your domains under, you will need to apply custom pages separately to each account.*

CLOUDFLARE®

1 888 99 flare | success@cloudflare.com | www.cloudflare.com