Smart Shield

- Get started
- Cloudflare Smart Shield
- Connection reuse
- Concepts
- Network diagram
- Argo Smart Routing
- Configuration
- Regional Tiered Cache
- Smart Tiered Cache
- Cache Reserve analytics
- Cache Reserve
- Cache Reserve operations
- Dedicated CDN Egress IPs
- IPs utilization
- Use with other Cloudflare products
- Setup
- Connection forwarding
- Egress IPs
- How it works
- Health Checks analytics
- Health Checks
- Manage Health Checks
- Zone Lockdown

Get started

import { GlossaryTooltip, DirectoryListing, Tabs, TabItem } from "~/components";

Smart Shield reduces the load on your origin server and improves content delivery by consolidating requests through Cloudflare's caching infrastructure. It is available to all customers as an opt-in configuration.

Before you begin

- You must have a Cloudflare account and onboard your domain.
- Verify that DNS records for the domain you want to protect are set to proxied. Smart Shield operates within Cloudflare's reverse proxy, so traffic from DNS-only records is not routed through it.

Steps

1. Log in to the Cloudflare dashboard, and select your account and domain.
2. Go to **Speed** > **Smart Shield**.
3. (Optional) Explore the different available packages.
4. Select **Get started for free** or choose a different package and select **Continue** to proceed to the guided onboarding flow.

After setup, you can monitor origin performance and cache effectiveness through the Observatory dashboard.

Packages and availability

Pro, Business, and Enterprise customers have access to Health Checks for monitoring origin availability across all packages.

Enterprise customers have access to all Smart Shield packages, including Smart Shield Advanced.

Smart Shield

The base package for reducing origin load through caching and connection optimization.

- Includes Smart Tiered Cache and Connection Reuse.

Smart Shield + Argo

Adds network path optimization on top of the base package. Use when visitors are geographically distant from the origin server.

- Includes Smart Tiered Cache, Connection Reuse, and Argo Smart Routing.

Smart Shield Advanced

The full package with additional caching customization through regional and persistent storage options.

- Includes Smart Tiered Cache, Connection Reuse, Argo Smart Routing, Regional Tiered Cache, and Cache Reserve.

Enterprise customers have access to Regional Tiered Cache as part of their plan, regardless of which Smart Shield package they use.

Enterprise customers also have the option to configure Dedicated CDN Egress IPs, allowing you to increase origin security by only allowing traffic from a small list of IP addresses. If you are interested, reach out to your account team.

Free, Pro, and Business customers can purchase Smart Shield and Smart Shield + Argo packages.

Smart Shield

The base package for reducing origin load through caching and connection optimization.

- Includes Smart Tiered Cache and Connection Reuse.

Smart Shield + Argo

Adds network path optimization on top of the base package. Use when visitors are geographically distant from the origin server.

- Includes Smart Tiered Cache, Connection Reuse, and Argo Smart Routing.

Smart Shield Advanced

Smart Shield Advanced is not currently available for Free, Pro, and Business customers. If you are interested in Smart Shield Advanced features such as Regional Tiered Cache and Cache Reserve, contact our Enterprise Sales team.

Further reading

Cloudflare Smart Shield

import { Description, Plan, RelatedProduct } from "~/components";

Protect your origin server, reduce load, and improve content delivery performance.
Every request that reaches your origin server costs resources — bandwidth, compute, and connections. When traffic spikes or your content is requested from many locations simultaneously, your origin can become a bottleneck. Smart Shield is a bundle of origin protection and performance features that reduce the number of requests and connections between Cloudflare's network and your origin server.

Smart Shield includes Smart Tiered Cache, which organizes Cloudflare data centers into upper-tier and lower-tier groups so that only upper-tier data centers contact your origin for uncached content.

Combined with connection reuse, which packages multiple requests into a single connection to your origin, Smart Shield reduces both the volume of origin requests and the number of open connections.

Depending on your package tier, Smart Shield can also include:

- Argo Smart Routing — routes traffic through the fastest network paths to reduce latency.
- Regional Tiered Cache — adds a regional cache layer between lower-tier and upper-tier data centers for geographic data locality (Enterprise plans, or Smart Shield Advanced).
- Cache Reserve — persistent cache storage that reduces cache misses for infrequently accessed content.
- Health Checks — monitors your origin server availability (Pro plans and above).
- Dedicated CDN Egress IPs — reserved IP addresses for origin allowlisting (Enterprise).

For a visual overview of how these features work together, refer to the network diagram.

Learn how to get started.

Related products

Cache stores copies of frequently accessed content (such as images, videos, or webpages) in geographically distributed data centers that are located closer to end users than origin servers, improving website performance. Observatory uses synthetic tests and real user data to assess the performance of your website, producing different metrics and insights. Cloudflare then uses this

analysis to recommend optimizations that best address your performance issues.

Connection reuse

Smart Shield reduces the number of connections between Cloudflare and your origin server by batching multiple requests through shared connections. When requests from an upper-tier data center — the layer of Cloudflare's cache that sits closest to your origin — need to reach your server, Smart Shield sends them over a single connection instead of opening a new connection for each request. This reduces overall connections to your origin by 30% on average, which lowers resource consumption on your origin and reduces the risk of connection exhaustion under high traffic.

For more information, refer to the Smart Shield announcement blog post.

About connection reuse

Every HTTP request requires a TCP connection between a client and a server. Each connection is identified by a pair of network addresses: the source IP address and port, and the destination IP address and port. Opening a new TCP connection has overhead — it requires a handshake between client and server, and a TLS negotiation if the connection is encrypted.

Connection reuse (also called persistent connections or keep-alive) avoids this overhead by sending multiple HTTP requests over a single TCP connection instead of opening a new connection for each request. HTTP/1.1 made this the default behavior.

For example, when a browser opens a connection to `shop.example.com`, the page may reference dozens of additional resources — stylesheets, images, scripts, and other files. Without connection reuse, each resource would require its own TCP connection. With connection reuse, all of these requests flow through the same connection.

Connection coalescing (HTTP/2)

With HTTP/2, connection reuse extends further through connection coalescing. This allows requests for different hostnames to share a single connection, as long as two conditions are met:

- The hostnames resolve to the same destination IP address and port.
- The TLS certificate on the server covers both hostnames (for example, a certificate that lists both `shop.example.com` and `blog.example.com` in its Subject Alternative Names).

This means a connection originally opened for `shop.example.com` can also carry requests for `blog.example.com`, reducing the total number of connections to your origin even further.

Connection reuse and Dedicated CDN Egress IPs

Connection reuse and connection coalescing are also considered when allocating your Dedicated CDN Egress IPs.

Concepts

import { DirectoryListing } from "~/components";

Network diagram

The diagram below shows how requests flow through the Cloudflare network when Smart Shield is active, and where each feature applies along that path.

Network diagram of requests being processed with all Smart Shield features

Requests from visitors first reach a nearby lower-tier data center. For static (cacheable) content, the lower-tier checks its local cache. On a cache miss, the request moves to an upper-tier data center — selected by Smart Tiered Cache based on lowest latency to your origin. If Regional Tiered Cache is configured, a regional hub is checked before the upper-tier. Persistent storage through Cache Reserve provides a final cache layer before requesting content from your origin.

For dynamic (non-cacheable) requests, Argo Smart Routing finds the fastest network path to your origin. Between Cloudflare's upper-tier data centers and your origin, connection reuse packages multiple requests into a single connection, reducing the total number of connections your origin handles.

Health Checks run from multiple data centers to monitor whether your origin is online and responsive. Dedicated CDN Egress IPs provide reserved IP addresses for traffic from Cloudflare to your origin, allowing you to restrict your origin firewall to a small allowlist.

Argo Smart Routing

import { Render } from "~/components";

:::note[Availability] Available with Smart Shield + Argo and Smart Shield Advanced. :::

Cloudflare provides analytics to show the performance benefits of Argo Smart Routing.

Configuration

import { DirectoryListing } from "~/components";

Regional Tiered Cache

import { Render } from "~/components";

:::note[Availability] Regional Tiered Cache is included with Enterprise plans. Smart Shield Advanced, which includes Regional Tiered Cache, is currently only available to Enterprise customers. If you are interested in Smart Shield Advanced, contact our Enterprise Sales team. :::

Smart Tiered Cache

import { Render } from "~/components";

:::note[Availability] Available in all Smart Shield packages. :::

With data centers around the world, Cloudflare caches content very close to end users. However, if a piece of content is not in cache, the Cloudflare data centers must contact the origin server to receive the cacheable content. Tiered cache works by dividing Cloudflare's data centers into a hierarchy of lower-tiers and upper-tiers, where only upper-tiers can ask your origin for content.

Cache Reserve analytics

import { Render } from "~/components";

Cache Reserve

import { Render } from "~/components";

:::note[Availability] Available with Smart Shield Advanced. :::

Smart Shield Advanced includes 2 TB of storage for Cache Reserve.

Asset eligibility

Limits

Delete data

You can remove all data stored in Cache Reserve. In most cases, deletion takes around 24 hours to be completed.

1. Select the three dots next to Cache Reserve in your Smart Shield configurations.
2. Choose **View details** to open the Cache Reserve sidebar.
3. Make sure to pause Cache Reserve.
4. Select **Delete data** and then **Save**.
5. Select **Delete** again in the dialog to confirm.

:::note If you want to purge your cache instead, refer to cache configurations. :::

Cache Reserve operations

import { Render } from "~/components";

Dedicated CDN Egress IPs

import { GlossaryTooltip, DirectoryListing, Render } from "~/components";

Enterprise customers can leverage dedicated egress[^1] IPs for layer 7 WAF and CDN services, as well as Spectrum. The egress IPs are reserved exclusively for your account so that you can increase your origin security by only allowing traffic from a small list of IP addresses.

:::note If you are interested in using Smart Shield Advanced with Dedicated CDN Egress IPs, reach out to your account team. :::

Dedicated CDN Egress IPs was formerly known as Cloudflare Aegis (release blog post).

Benefits

With Dedicated CDN Egress IPs, you can:

- Lock down your network firewall to only allow traffic from your dedicated IPs.
- Use Cloudflare Access and CNI to secure your applications without installing software or customizing code on your server.
- Ensure only authorized Workers can access your origin services.

Scope

You can assign Dedicated CDN Egress IPs to single or multiple Cloudflare zones, and across different Cloudflare accounts.

Dedicated CDN Egress IPs are included within BGP advertisement over CNI.

Resources

[^1]: From Cloudflare to your origin. Refer to how it works for details.

IPs utilization

import { Type, Render } from "~/components";

Use the GraphQL API to get aggregate data and monitor your dedicated IPs capacity (formerly known as Aegis).

Refer to the GraphQL Analytics API documentation for further guidance, or consider the example below for a quickstart.

GraphQL schema

The specific schema to get Dedicated CDN Egress IPs data is called `aegisIpUtilizationAdaptiveGroups`.

You can get average (`avg`) or maximum (`max`) utilization values (in percentage), and use the following dimensions:

- `datetimeFiveMinutes`
  - Timestamp truncated to five minutes. For example, `2025-01-10T00:05:00Z`.
- `popName`
  - The Cloudflare point of presence (PoP). For example, `sjc`.
- `egressIp`
  - Your assigned Dedicated CDN Egress IP. For example, `192.0.2.1`.
- `origin`
  - Origin IP and port. For example, `203.0.113.150:443`.
- `popUtilizationKey`
  - The Cloudflare point of presence (PoP), the Dedicated CDN Egress IP, and the origin IP and port. For example, `sjc 192.0.2.1 203.0.113.150:443`.

Example

Refer to the query below to learn how to get average utilization and maximum utilization by point of presence, and filter the results.

You can also select the button at the bottom to use this query for your account via the Cloudflare GraphQL API Explorer. Make sure to provide your account ID and timestamps, and replace the

placeholders for `popName`, `egressIp`, and `origin` as needed.

```
query AegisIpUtilizationQuery(
  $accountTag: string
  $datetimeStart: string
  $datetimeEnd: string
) {
  viewer {
    utilization: accounts(filter: { accountTag: $accountTag }) {
      avgByPopUtilization: aegisIpUtilizationAdaptiveGroups(
        limit: 100
        filter: {
          datetimeFiveMinutes_geq: $datetimeStart
          datetimeFiveMinutes_leq: $datetimeEnd
        }
        orderBy: [datetimeFiveMinutes_ASC]
      ) {
        avg {
          utilization
        }
        dimensions {
          datetimeFiveMinutes
          popUtilizationKey
        }
      }

      maxByPopUtilization: aegisIpUtilizationAdaptiveGroups(
        limit: 100
        filter: {
          datetimeFiveMinutes_geq: $datetimeStart
          datetimeFiveMinutes_leq: $datetimeEnd
        }
        orderBy: [datetimeFiveMinutes_ASC]
      ) {
        max {
          utilization
        }
        dimensions {
          datetimeFiveMinutes
          popUtilizationKey
```

```
    }
  }

  filterPopUtilization: aegisIpUtilizationAdaptiveGroups(
    limit: 100
    filter: {
      datetimeFiveMinutes_geq: $datetimeStart
      datetimeFiveMinutes_leq: $datetimeEnd
      popName: "<CLOUDFLARE_POP>"
    }
    orderBy: [datetimeFiveMinutes_ASC]
  ) {
    max {
      utilization
    }
    dimensions {
      datetimeFiveMinutes
      popUtilizationKey
    }
  }

  filterIPUtilization: aegisIpUtilizationAdaptiveGroups(
    limit: 100
    filter: {
      datetimeFiveMinutes_geq: $datetimeStart
      datetimeFiveMinutes_leq: $datetimeEnd
      egressIp: "<YOUR_EGRESS_IP>"
    }
    orderBy: [datetimeFiveMinutes_ASC]
  ) {
    max {
      utilization
    }
    dimensions {
      datetimeFiveMinutes
      popUtilizationKey
    }
  }

  filterOriginUtilization: aegisIpUtilizationAdaptiveGroups(
```

```
        limit: 100
        filter: {
          datetimeFiveMinutes_geq: $datetimeStart
          datetimeFiveMinutes_leq: $datetimeEnd
          origin: "<ORIGIN_IP_AND_PORT>"
        }
        orderBy: [datetimeFiveMinutes_ASC]
      ) {
        max {
          utilization
        }
        dimensions {
          datetimeFiveMinutes
          popUtilizationKey
        }
      }
    }
  }
}
```

Use with other Cloudflare products

Use Dedicated CDN Egress IPs in combination with different Cloudflare products.

Access and CNI

You can use Dedicated CDN Egress IPs combined with Cloudflare Network Interconnect (CNI) to secure your applications with Cloudflare Access without installing software or customizing code on your server.

While Access allows you to enforce policies at the hostname level, other solutions are usually necessary to protect against origin IP bypass [^1]. With Dedicated CDN Egress IPs, you only allow a small number of IPs (that are not publicly listed) through your network firewall and, with Cloudflare Network Interconnect, you can use a completely private path between Cloudflare and your application server, without exposure to the public Internet. For details and background, refer to the Cloudflare blog.

Dedicated CDN Egress IPs are included within BGP advertisement over CNI.

Data Localization Suite

Data Localization Suite (DLS) is an enterprise add-on that enables you to choose the location where Cloudflare encrypts, decrypts, and stores data.

To ensure egress will happen from DLS-specified locations, make sure you have Dedicated CDN Egress IPs provisioned in those locations. Refer to IPs allocation for details.

Load Balancing

Cloudflare Load Balancing allows you to intelligently distribute traffic across your origins by issuing regular monitors (that assess origin health) and following the traffic steering policies you define.

By default, the Load Balancing monitors will use public Cloudflare IP addresses.

To avoid inconsistencies between what the Load Balancing monitors report and what you observe in service traffic with Dedicated CDN Egress IPs, make sure to turn on the **Simulate Zone** option in the monitor settings.

Spectrum

Spectrum allows you to route email, file transfer, games, and more over TCP or UDP through Cloudflare. This means you can mask your origin and protect it from DDoS attacks.

While you can use BYOIP or static IPs to control which IPs are used for ingress with Spectrum, Dedicated CDN Egress IPs allows you to have a more strict list of egress IPs as well.

Dedicated CDN Egress IPs with Spectrum supports both TCP and UDP application types. HTTP/HTTPS types are also supported, although through a different configuration.

If you are interested in any of these solutions, contact your account team.

Workers

Workers provides a serverless execution environment for you to create applications leveraging Cloudflare's global network.

Refer to the sections below for information on how Dedicated CDN Egress IPs pair up with Workers.

`fetch`

`fetch()` requests that access services on your origin will use Dedicated CDN Egress IP addresses.

Workers subrequests — requests from one Worker to another — are expected to use different IPs. However, `fetch()` requests to external origins made by a Worker invoked via a subrequest will use Dedicated CDN Egress IP addresses.

`connect`

For `connect()` requests - which create outbound TCP connections from Workers - Dedicated CDN Egress IPs are **not** used.

[^1]: When an attacker knows your origin server IP and uses it to directly interact with the target application.

Setup

import { APIRequest, Render } from "~/components";

You can use the Edit Zone Settings API endpoint to set up Dedicated CDN Egress IPs (formerly known as Aegis). If you are not familiar with how Cloudflare API works, refer to Fundamentals.

:::note[Enterprise-only] Dedicated CDN Egress IPs (DCEI) are currently available to Enterprise customers. Contact your account team to request access. :::

Requirements

- The Dedicated CDN Egress IPs (DCEI) zone setting is only available within Cloudflare accounts that own leased IPs, or accounts to which a BYOIP prefix has been delegated. If you wish to use Dedicated CDN Egress IPs for zones that do not meet this criteria, contact your account team.
-

:::caution You must allowlist the IP addresses from this pool in your infrastructure before linking it to a zone. If you skip this step, traffic will not reach your origin, causing errors. :::

Turn on DCEI for a zone

1. Contact your account team to get the ID for your dedicated egress pool.
2. Make a `PATCH` request to the Edit Zone Setting endpoint:

- Specify `aegis` as the setting ID in the URL.
- In the request body, set `enabled` to `true` and use the ID from the previous step as the `pool_id` value.

<APIRequest path="/zones/{zone_id}/settings/{setting_id}" method="PATCH" json={{ id: "aegis", value: { enabled: true, pool_id: "<EGRESS_POOL_ID>", }, }} parameters={{ setting_id: "aegis", }} />

Check DCEI status for a zone

<APIRequest path="/zones/{zone_id}/settings/{setting_id}" method="GET" parameters={{ setting_id: "aegis", }} />

Turn off DCEI for a zone

<APIRequest path="/zones/{zone_id}/settings/{setting_id}" method="PATCH" json={{ id: "aegis", value: { enabled: false, }, }} parameters={{ setting_id: "aegis", }} />

Check your IPs

<Render file="check-leased-ips" product="byoip" params={{ product: "dedicated IPs for CDN egress", }} />

If you are using BYOIP, refer to **BYOIP prefixes** instead.

Connection forwarding

Since IPv6 address ranges are deployed globally, no forwarding is needed.

For IPv4 traffic, based on IPs allocation, not all egress data centers will have access to an applicable dedicated CDN egress IP.

Dedicated CDN egress IPs do not forward to another location in response to traffic spikes. Instead, each IPv4 can be split across up to four locations, where some of these locations may have multiple data centers. IP capacity in each data center can also be adjusted in accordance with the amount of traffic that reaches each location.

After a request reaches Cloudflare on an ingress data center, and the cache service sends a request for the egress router to connect to your origin, the following scenarios are possible.

Traffic can egress from the same server

If the server running the egress router has access to an applicable dedicated CDN egress IP, traffic egresses from that server.

```
flowchart LR
        accTitle: Dedicated CDN Egress IPs and connection forwarding
        accDescr: Diagram showing IPv4 connection forwarding for Dedicated CDN Egress IPs -
Same data center.
        A[Client]
        subgraph Data center A
        X[(Cache service)] --> B[(Egress router <br/> <small>*has applicable IP</small>)]
        end
        C[(Origin server)]

        A --ingress--> X
        B --egress--> C
```

Connection forwarding is needed

If the server does not have access to an applicable IP, the following options are checked and the first that is possible will take place:

- Another server in the same data center has access to an applicable IP and the connection is forwarded to that server.

```
flowchart LR
        accTitle: Dedicated CDN Egress IPs and connection forwarding
        accDescr: Diagram showing IPv4 connection forwarding for Dedicated CDN Egress IPs -
Same data center.
        A[Client]
        subgraph Data center A
        X[(Cache service)] --> B[(Egress router <br/> <small>*no applicable IP</small>)]
        B --> Y[(Egress server <br/> <small>*has applicable IP</small>)]
        end
        C[(Origin server)]

        A --ingress--> X
        Y --egress--> C
```

- Another data center in the same location has access to an applicable IP and the connection is forwarded to that data center.

```
flowchart LR
        accTitle: Dedicated CDN Egress IPs and connection forwarding
        accDescr: Diagram showing IPv4 connection forwarding for Dedicated CDN Egress IPs -
Different data center.
        A[Client]
        subgraph Location 1
        subgraph Data center A
        X[(Cache service)] --> B[(Egress router <br/> <small>*no applicable IP</small>)]
        end
        subgraph Data center B
        B --> Y[(Egress server <br/> <small>*has applicable IP</small>)]
        end
        end
        C[(Origin server)]


        A --ingress--> X
        Y --egress--> C
```

- Another data center in a different location has access to an applicable IP. The closest location is selected and connection is forwarded to that location.

```
flowchart LR
        accTitle: Dedicated CDN Egress IPs and connection forwarding
        accDescr: Diagram showing IPv4 connection forwarding for Dedicated CDN Egress IPs -
Different location.
        A[Client]
        subgraph Location 1
          subgraph Data center A
          X[(Cache service)] --> B[(Egress router <br/> <small>*no applicable IP</small>)]
          end
        end
        subgraph Location 2
          subgraph Data center C
            B --> Y[(Egress server <br/> <small>*has applicable IP</small>)]
          end
        end
        C[(Origin server)]


        A --ingress--> X
        Y --egress--> C
```

Egress IPs

import { Render } from "~/components";

When you use Cloudflare as a reverse proxy, Cloudflare's global network sits between client requests and your origin servers.

```
flowchart LR
        accTitle: Cloudflare as a reverse proxy
        accDescr: Diagram showing Cloudflare's network between clients and the origin server.
        A[Client] <--> B((Cloudflare))<--> C[(Origin server)]
```

Zooming into what happens as a request routes through Cloudflare, you can consider two parts of the process: ingress and egress.

```
flowchart LR
        accTitle: Cloudflare as a reverse proxy
        accDescr: Diagram showing Cloudflare's network between clients and the origin server.
        A[Client] --ingress--> B((Cloudflare))--egress--> C[(Origin server)]
```

Ingress refers to the data center where the client request lands on, based on Internet routing. From there on, the request will be processed according to your Cloudflare configurations and, if needed, a connection to the origin will be initiated via an egress data center.

Traditionally, Cloudflare maintains a very large pool of egress IPs that are used by all Cloudflare customers and are publicly documented. With Dedicated CDN Egress IPs, Cloudflare connects to your origin using IPs that are reserved for you.

BYOIP or Cloudflare-leased

Each dedicated CDN egress IP pool can consist of either IPs from a BYOIP prefix or Cloudflare-leased IPs. A single dedicated CDN egress IP pool cannot contain both BYOIPs and leased IPs.

<Render file="check-leased-ips" product="byoip" params={{ product: "dedicated IPs for CDN egress", }} />

If you are using BYOIP, refer to **BYOIP prefixes** instead.

IPs allocation

Dedicated CDN Egress IPs support both IPv4 and IPv6 addresses.

IPv6 address ranges are deployed globally, meaning your dedicated IPv6 addresses can be used for connections from Cloudflare to your origin servers across all Cloudflare data centers.

:::note[China exception] Dedicated CDN Egress IPs are currently **not** available in the Cloudflare China Network. :::

For IPv4 addresses, you should work with your account team to choose the locations where each IP should be deployed. Ideally, your dedicated IPv4 addresses should be placed near your origin servers and adjusted to the amount of traffic expected for each region.

Refer to connection forwarding to understand how requests are processed when reaching different Cloudflare data centers.

Connections to your origin

Dedicated CDN Egress IPs also benefit from connection reuse and coalescing.

GraphQL Analytics API allows you to get visibility over IPs utilization.

Regional Services

If you are using Regional Services, you should take this into consideration when allocating dedicated IPv4 addresses. Traffic will egress from the specified locations as long as you have Dedicated CDN Egress IPs provisioned in those locations.

How it works

import { DirectoryListing } from "~/components";

Refer to the pages below to understand how Dedicated CDN Egress IPs work.

Health Checks analytics

import { Render } from "~/components";

Use Health Checks analytics to debug possible origin issues.

Health Checks

import { Render, DirectoryListing } from "~/components";

:::note[Availability] Available for Pro, Business, and Enterprise customers in all Smart Shield packages. :::

Regions

Further reading

Manage Health Checks

import { AvailableNotifications, Render } from "~/components";

Refer to the section below to learn how to manage your Smart Shield health checks.

Create and edit health checks

1. Log in to the Cloudflare dashboard and select your account and domain.
2. Go to **Speed** > **Smart Shield**.
3. For Health Checks, select **Manage**.
4. Select **Create** or find an existing health check and select **Edit**.
5. Fill out the form or edit existing values, paying special attention to:
   - The values for **Interval** and **Check regions**, because decreasing the **Interval** and increasing **Check regions** may increase the load on your origin server.
   - **Retries**, which specify the number of retries to attempt in case of a timeout before marking the origin as unhealthy.
6. Select **Save and Deploy**.

Configure alerts

You can configure notification emails to be alerted when the health check detects that there is a change in the status of your origin server. Cloudflare will send you an email within seconds so you can take the necessary action before customers are impacted.

The email provides information to determine what caused the health status change. You can evaluate when the change happened, the status of the origin server, if and why it is unhealthy, the expected response code, and the received response code. Refer to common error codes for further guidance.

1. Log in to the Cloudflare dashboard and select your account and domain.
2. Go to **Speed** > **Smart Shield**.
3. For Health Checks, select **Manage** and then **Configure an alert**.
4. Fill out the **Notification name** and **Description**.
5. Add a Notification email.
6. Select **Next**.
7. Add health checks to include in your alerts.

8. Choose the **Notification trigger**, which determines when you receive alerts.
9. Select **Create**.

Zone Lockdown

import { Render } from "~/components";