

Overview	2
Cloud based security and performance providers	2
Cloudflare's reverse proxy architecture and solution	4
Cloudflare onboarding options	5
Why multi-vendor?	7
Regulatory/company compliance	7
Resiliency	7
Performance	8
Cost	8
Multi-vendor solution considerations	8
Routing	9
Configuration	10
Connectivity	10
Common deployments	14
Multi-vendor active-active security and different provider for DNS	14
Multi-vendor active-active security with multi-vendor DNS from same providers	15
Multi-vendor DNS setup options	17
Configuration and management best practices	20
Connectivity options	21
Internet (default)	21
Private connection - tunnel or VPN	22
Direct connection	23
Additional routing and security options	23
Argo Smart Routing	24
Authenticated Origin Pulls (mTLS)	24
Summary	25

Overview

Over time and with the rapidly evolving application security and performance industries, companies have come to deploy multiple vendors to provide services. Sometimes customers opt for using multiple vendors for reasons of regulatory/company compliance, resiliency, performance, or cost.

Although some customers look to implement multi-vendor solutions for various reasons discussed in this document, multi-vendor deployments can introduce additional complexity, higher operational costs due to multiple dashboards and configurations, and a steeper learning curve. Additionally, while trying to establish a baseline of supported features across multiple vendors, customers can end up having a minimum common denominator setup, not taking advantage of the latest capabilities/innovations from a vendor. Customers should carefully consider the goals and requirements, and weigh pros and cons with all stakeholders, before proceeding with a multi-vendor deployment.

This document examines why some customers deploy a multiple or dual vendor approach and how Cloudflare can be incorporated into such a solution. Specifically, this document describes how a multi-vendor approach for application security and performance can be accomplished. This document is targeted for architects and those interested in using multi-vendor cloud-based solutions for security and performance.

Cloud based security and performance providers

Before discussing multi-vendor security and performance solutions, it's important to note how cloud-based solutions providing these services work in general and how traffic is routed through them.

Cloud-based security and performance providers like Cloudflare work as a reverse proxy. A reverse proxy is a server that sits in front of web servers and forwards client requests to those web servers. Reverse proxies are typically implemented to help increase security, performance, and reliability.

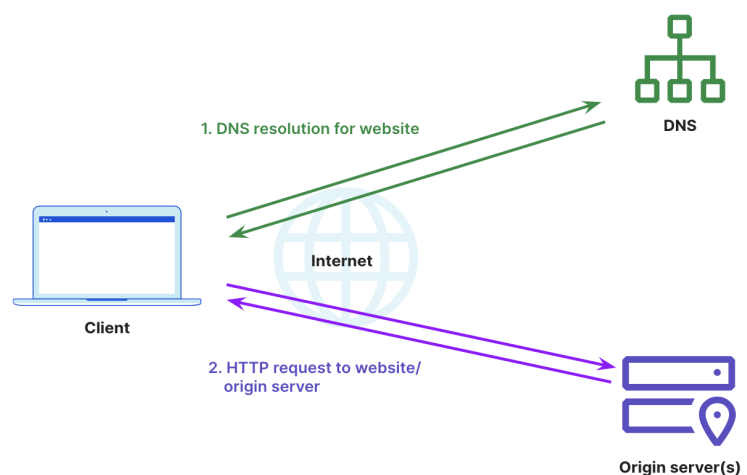


Figure 1: Client request to origin server

Normal traffic flow without a reverse proxy would involve a client sending a DNS lookup request, receiving the origin IP address, and communicating directly to the origin server(s). This is visualized in Figure 1.

When a reverse proxy is introduced, the client still sends a DNS lookup request to its resolver, which is the first stop in the DNS lookup. In this case, the DNS resolver returns a vendor's reverse proxy IP address to the client and the client then makes a request to the vendor's reverse proxy. The cloud-based proxy solution can now provide additional security, performance, and reliability services like [CDN](#), [WAF](#), [DDoS](#), [API Gateway](#), [Bot Management](#) capabilities, etc, before deciding, based on security policy, whether to route the client request to the respective origin server(s). This is visualized in Figure 2.

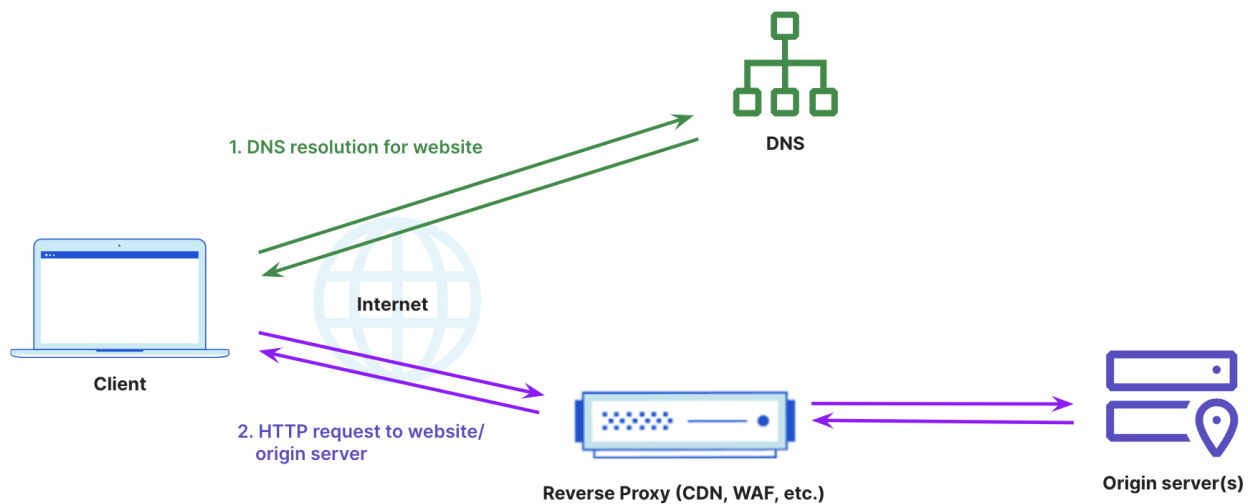


Figure 2: Client request routed through reverse proxy for additional security and performance services

In some cases, the vendor providing the reverse proxy also provides DNS services; this is visualized in Figure 3 below. This can be beneficial for managing all services from a single dashboard and for operational simplicity.

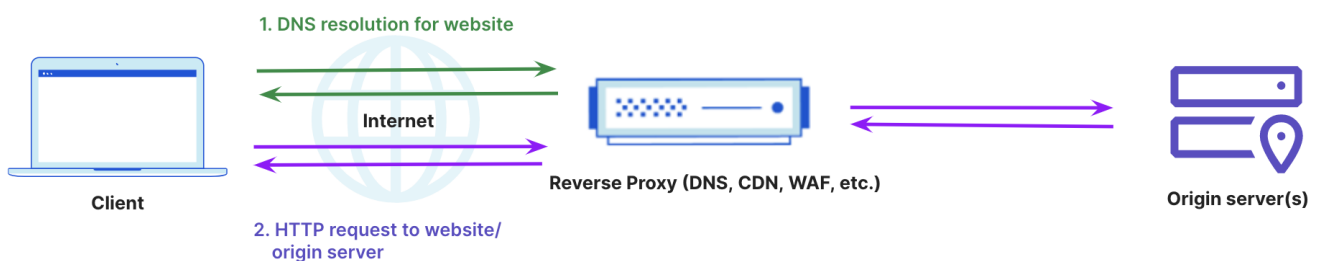


Figure 3: Same vendor providing DNS and security/performance services via proxy

Cloudflare's reverse proxy architecture and solution

Cloudflare provides a reverse proxy architecture using its global [Anycast network](#) for the respective security, performance, and reliability services it provides. Anycast is a network addressing and routing method in which incoming requests can be routed to a variety of different locations or 'nodes' advertising the same IP address space. Cloudflare is extremely performant and reliable thanks to Anycast, as well as its global presence in over 300 cities across 100+ countries. Cloudflare is also directly connected to 12,000 networks, including every major ISP, cloud provider, and enterprise, and within ~50 ms from 95% of the world's Internet-connected population.

Cloudflare has one global network with every service running on every server in every Cloudflare data center. Since Cloudflare's network uses Anycast, the closest data center to the client will respond to the client request. This decreases latency while improving network resiliency, availability, and security due to the increased overall distribution of traffic across Cloudflare's network.

[Cloudflare's Global Anycast Network](#) provides the following advantages:

- Incoming traffic is routed to the nearest data center with the capacity to process the requests efficiently.
- Availability and redundancy is inherently provided. Since multiple nodes advertise the same IP address, if one node were to fail, requests are simply routed to another node in close proximity.
- Because Anycast distributes traffic across multiple data centers, it increases overall distribution of traffic across Cloudflare's network, preventing any one location from becoming overwhelmed with requests. For this reason, Anycast networks are very resilient to DDoS attacks.

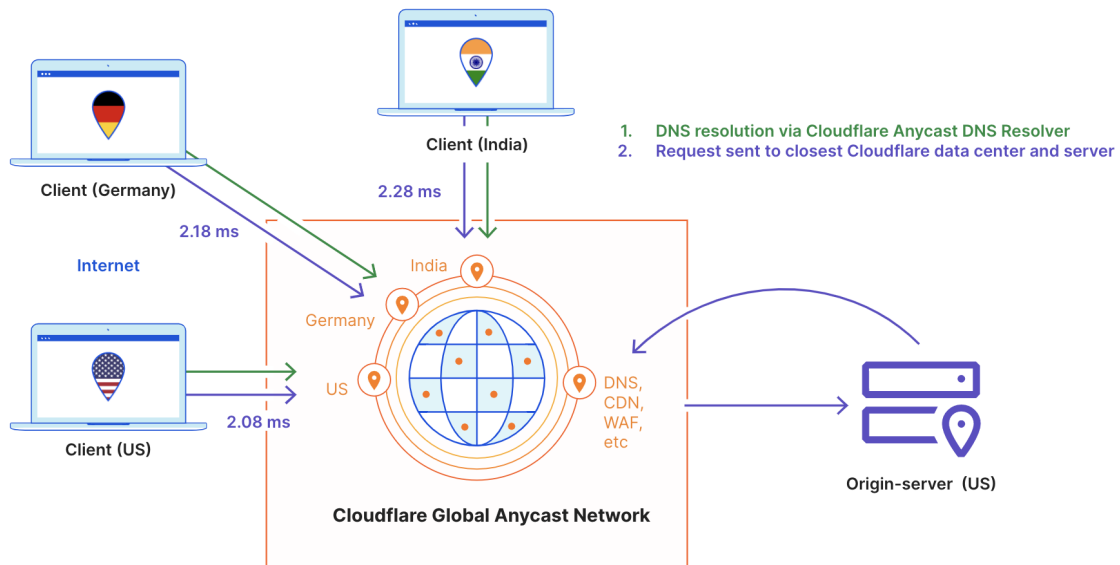


Figure 4: Cloudflare providing DNS and security/performance services via Global Anycast Network

Cloudflare onboarding options

This section provides a brief overview of the Cloudflare onboarding options which are useful to understand prior to looking into the details around a multi-vendor solution. The method of onboarding allows for variance in how the multi-vendor solution is deployed/configured. If you're already familiar with the Cloudflare onboarding options, you can jump to the next section discussing multi-vendor solutions.

Cloudflare provides multiple options to easily onboard and consume security, performance, and reliability services. One of the advantages of cloud solutions offered via proxy setup is the ease of onboarding and getting started because it primarily involves DNS configuration to route client requests through the proxy. However, even within the onboarding with DNS configuration, Cloudflare offers multiple options and flexibility.

The core requirement is, traffic must be proxied through Cloudflare; this is also referred to as 'orange-clouded,' because the traffic to the site is being proxied through Cloudflare. Within the dashboard, you will see the status for a specific DNS entry as 'Proxied' and the orange cloud icon as shown in Figure 5 below.

DNS management for **cf-tme.com** Import and Export ▾ ⚙ Dashboard Display Settings

All changes made in the edit drawer are implemented once saved.

Search DNS Records

▽ Add filter Search + Add record

Type	Name	
LB	lb.cf-tme.com	Manage in Traffic app
Spectrum	splunk.cf-tme.com	Manage in Spectrum app


Type ▲	Name	Content	Proxy status	TTL	Actions
A	api2	35.212.251.66	 Proxied	Auto	Edit ▶

Figure 5: Cloudflare configured to proxy traffic for site https://api2.cf-tme.com

There are several methods to proxy traffic through Cloudflare and the method used will depend on customer requirements.

1. Full DNS setup - Cloudflare as primary DNS provider

Cloudflare is configured as the primary DNS provider and A records are configured to proxy traffic through Cloudflare. When the proxy is enabled on a DNS record, the response will be Cloudflare Anycast IP addresses allowing for Cloudflare to be the proxy.

2. Secondary DNS setup with Secondary DNS override

Cloudflare is configured as a secondary provider and all DNS records are transferred from the primary provider. Cloudflare provides a feature called [Secondary DNS override](#) that allows customers to override the response served from Cloudflare secondary nameservers. This allows for customers to take advantage of leveraging zone transfers to automatically sync between DNS providers. It also provides the flexibility to update select records in Cloudflare DNS to redirect certain traffic to another service provider like Cloudflare. In this case, the response will be Cloudflare Anycast IP addresses allowing for Cloudflare to be the proxy.

3. Partial / CNAME setup

In this setup, Cloudflare is not the authoritative DNS provider and the customer manages DNS records externally.

Converting to CNAME setup ensures the hostname eventually resolves to Cloudflare IPs. This is useful when customers don't want to change their current DNS setup but still want to use other Cloudflare services.

If a customer's current DNS provider doesn't support CNAME on the zone apex (sometimes called the "root domain" or "naked domain") like Cloudflare does with [CNAME Flattening](#), you must purchase Static IPs from Cloudflare and create an A record to those Static IPs in the provider DNS. In Cloudflare, you can then create an A record to point the zone apex to the origin.

Many customers using Cloudflare services take advantage of the cross-product integration and innovations along with simplicity of a single UI for management and operational simplicity and use multiple Cloudflare services together like CDN and WAF. Although not recommended, it's also possible to use security services like WAF with other CDN providers by setting up DNS to forward traffic through Cloudflare via CNAME and disabling Cloudflare caching via Cache Rules.

Why multi-vendor?

Typically customers opt for a multi-vendor approach for reasons of regulatory/company compliance, resiliency, performance, and cost.

Regulatory/company compliance

Some customers may have to comply with regulatory/company policy of not being dependent on a single vendor for all security, performance, and reliability services. This could be done for reasons of a company's policy of mitigating risk for specific vendor outages/issues and/or for leverage to mitigate against increased vendor pricing/costs. For compliance with these policies, a multi-vendor strategy is required.

Resiliency

When a single vendor is used for all security and performance services, this may be perceived as a single point of failure. This can be driven by regulatory pressure to improve reliability in all critical systems, outages experienced with an incumbent vendor, or uncertainty with the long term reliability of a single vendor.

Performance

In many cases a single vendor may be very well connected and provide the expected level of performance within a certain region, but less so in other regions; this could be due to a number of reasons including investment, limited resources, geopolitical reasons, etc. Many customers desire to fully optimize speed in performance critical applications and media by implementing a multi-vendor approach that is often coupled with real time performance monitoring to steer traffic to the most optimal vendor based on that data.

Cost

Just like the performance of a particular vendor can vary based on content, time of day, and location, so can the cost, and sending particular traffic through a particular vendor can help optimize the overall cost of the delivery. Typically these benefits are seen driving a multi-vendor strategy in very specific use cases, such as for high volume media traffic, as the cost of onboarding and managing multiple vendors typically increases monetary and resource costs outside of specific niche use cases. Additionally, adopting a multi-vendor approach helps avoid vendor lock-in with any single provider, offering greater flexibility and negotiating power across vendors.

Multi-vendor solution considerations

Any multi-vendor architecture will contain several components an organization must decide on prior to implementing, both on the business and technical side. Additionally, there are several things to keep in mind to help optimize your setup to align with Cloudflare's strengths and unique differentiators.

Optimize for feature set and delivery methodology. Cloudflare is able to offer feature parity with most major vendors, with custom features easily delivered through our serverless compute service. For delivery methodology, Cloudflare's Anycast architecture is unique in that every server can deliver every service that Cloudflare offers, making it an optimal candidate for an active/active approach.

Leverage Cloudflare's API and rapid deployment capabilities wherever possible. Since Cloudflare offers every feature API first, and config changes typically are visible in a few seconds, this makes it easy for teams to test and deploy changes in a programmatic fashion without having to wait for long deployment times.

Avoid a “stacked” approach. This means avoid having Cloudflare placed in the request flow behind another vendor. We often hear companies consider stacking vendors with the hope of providing defense in depth by running the same traffic through each layer in a linear fashion. In theory this would allow for both vendors' policies to be run, and any bad traffic not caught by one vendor is hopefully caught by the next. What we see in practice when this setup is used is very different. The main disadvantage is the loss of full traffic visibility when sitting behind another vendor, which hinders many of Cloudflare’s threat intelligence powered services such as Bot Management, Rate Limiting, DDoS mitigation, & IP reputation database. This is also highly suboptimal from the performance side since the traffic must pass through two networks each with their own processing and connection overhead before going back to origin. Also, it creates unnecessary complexity in operations, management, and support.

One note on a stacked approach is that in certain cases for particular point solutions, it can make sense to place one vendor solution in front of the other, such as particular bot management solutions and API gateways, especially when migrating towards a new vendor/provider. In these scenarios it’s important to understand where each solution falls in the request flow to optimize effectiveness.

While Cloudflare and many providers maintain a high degree of availability and a robust fault tolerant architecture, some customers have a further desire to reduce dependency and respectively single vendor point of failures. It’s important to plan for a worst case scenario where some or all of a vendor's services are down and how to work around that in a short timeframe. Customers must consider how to have redundancy across DNS providers, networks, and origin connectivity to eliminate the risk of a single vendor/component failure cascading into a widespread outage.

While the specifics may vary widely depending on the vendor and business case, the technical considerations for a multi-vendor deployment can be bucketed into three areas: routing logic, configuration management and origin connectivity.

Routing

The first and likely most important decision that must be made when looking at a multi-vendor strategy is how to route traffic to each provider. This depends on both the business logic driving the multi-vendor strategy and the technical capabilities of each vendor in question. Traffic to each provider will be routed using DNS and shift depending on the current conditions and needs of the business. Cloudflare can support configurations as an authoritative DNS provider, secondary DNS provider, or non-Cloudflare DNS (CNAME) setups for a zone.

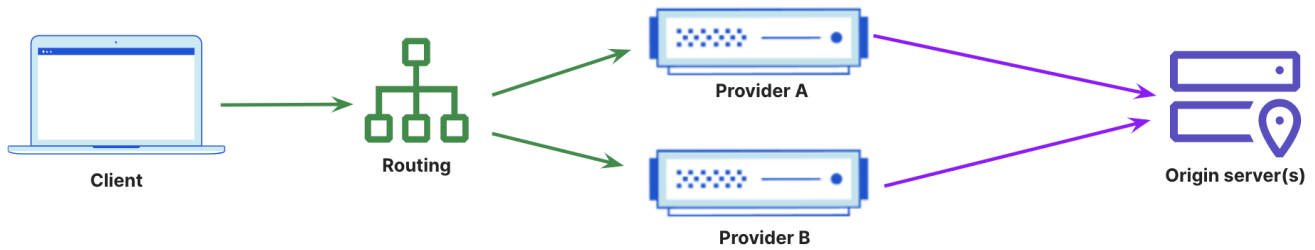


Figure 6: Client request being routed to origin server(s) in a multi-vendor setup

DNS based load balancing and health checks can be leveraged here so that client requests to the domain/site are distributed across healthy origin server(s). The DNS provider monitors the health of the servers and DNS responds to the client request using a round-robin approach with the respective IPs.

If a multi-vendor DNS approach is also desired for DNS-level resiliency, a variety of configurations are possible with multiple authoritative nameservers from different vendors. See the 'Multi-vendor DNS setup options' section in this document for additional details. The key here is ensuring consistent configurations across multiple providers. Depending on the DNS setup/configuration, this consistency can be resolved using different approaches such as zone transfers, automation via tools such as Terraform or OctoDNS, monitoring/automation via scripting, or even manual configuration.

Configuration

While many vendors can deliver a similar end user experience, configuration and management can differ greatly between providers, which drives up the cost of a successful implementation. Ultimately that means the business must become familiar with each vendor's configuration logic and develop a system to map between them. Wherever possible, seek out vendors that optimize for management simplicity, automation support, and rapid deployment to help minimize the cost and management overhead.

API support for all vendor's product functionality becomes critical here. Maintaining consistent configuration is important not only in the routing in certain multi-vendor DNS setups but also for maintaining consistency between all of the respective services such as WAF, API security, etc. as traffic can be routed to either provider. Automation tools such as Terraform or custom scripted automation tools will leverage the APIs to maintain this consistency between vendors.

Connectivity

Another important decision that must be made is how each provider will connect back into your organization. This will largely depend on the vendor's capabilities plus the technical and security

requirements of your organization.

Clients will make requests over the Internet and the requests will be routed to the respective vendor's proxy service on the vendor's cloud. In the most basic scenario, the proxy will simply route the traffic over the Internet to the origin; this is the default setup.

If the customer wants more security or additional performance benefits, they may decide to also leverage vendor offered connectivity options such as encrypted tunnels to origin or direct connect options from customer data centers directly to Cloudflare data centers via cross connect from a customer's equipment to Cloudflare. Vendors may also offer accelerated routing capabilities where they actively monitor the fastest paths over the Internet to ensure the most optimal routes to the origin are used.

Cloudflare offers all of these connectivity options along with Smart Routing to ensure the fastest paths to origin are used. These connectivity options are discussed in more detail in the 'Cloudflare connectivity options' section of this document.

Operations and Troubleshooting

Some important considerations when designing a multi-vendor solution are operations and troubleshooting. Having a multi-vendor solution can raise operational costs and also impact troubleshooting as you now have two different environments to manage and troubleshoot.

A primary focus for Cloudflare has always been operational simplicity and providing visibility. Cloudflare provides a single unified dashboard where all security, performance, and reliability services can be accessed from a consistent operationally simple UI.

Additionally, Cloudflare offers logging, analytics and security analytics dashboards. Logs with additional details are also accessible from the UI. Customers have granular data that can be used for analysis and troubleshooting.

Figure 7 below shows a view of Cloudflare Security Analytics which brings together all of Cloudflare's detection capabilities in one place. This provides security engineers and admins with a quick view of current traffic and security insights in regards to their site.

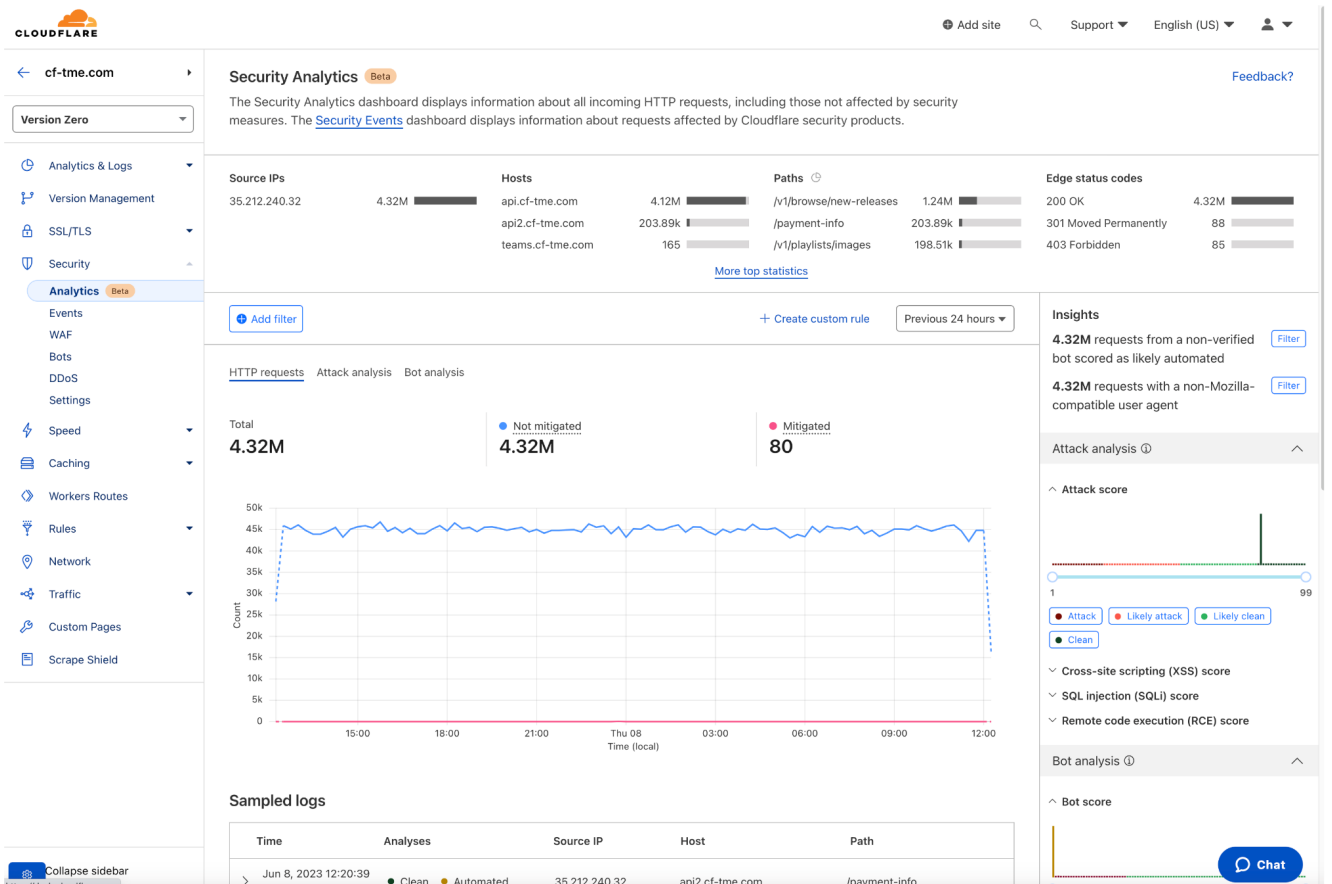
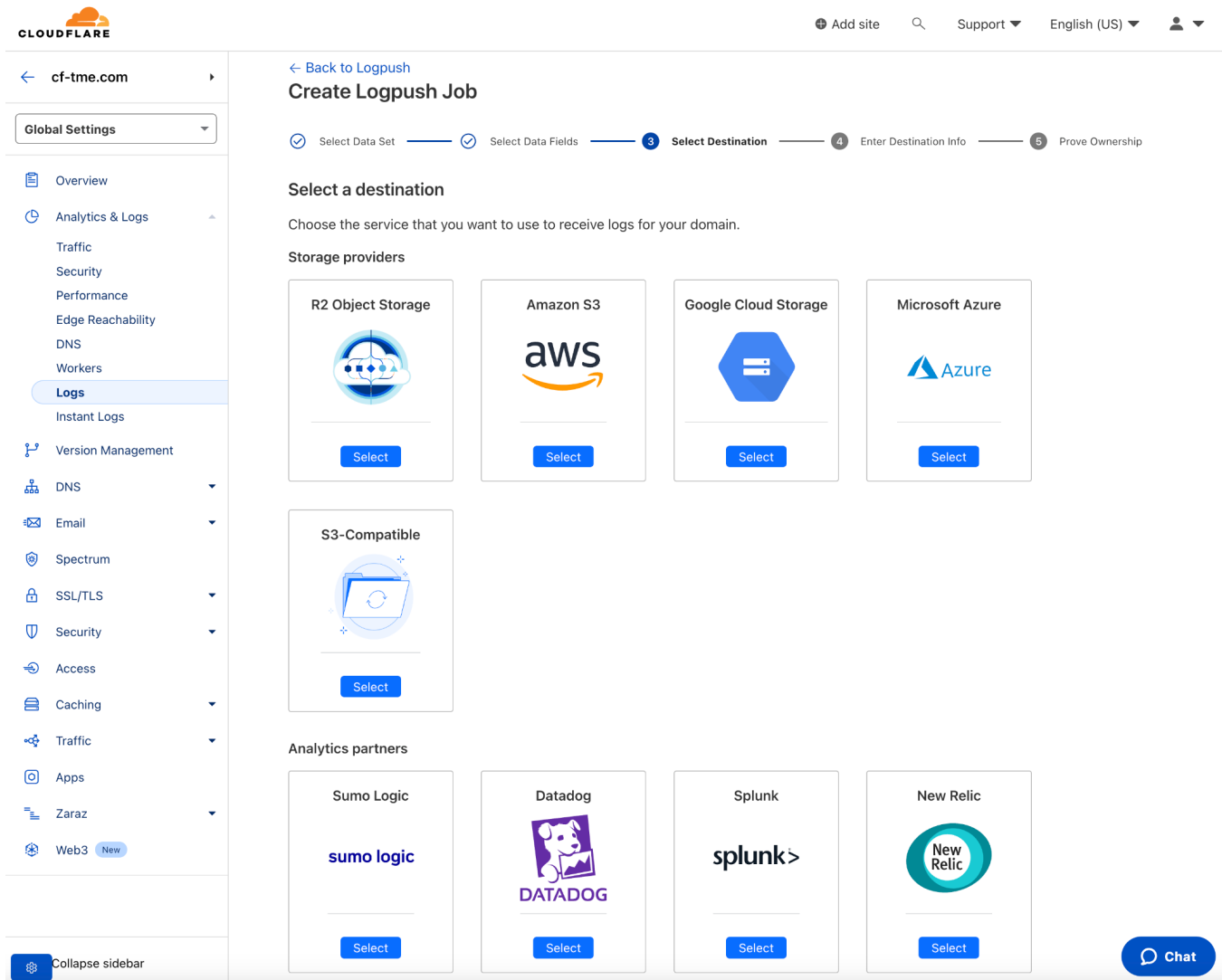


Figure 7: Cloudflare Security Analytics

In addition to analytics for each product and security analytics shown above, you can also view logs within the UI and export logs to Cloudflare or third party clouds or products for additional analysis.

In Figure 8 below a Logpush is being configured to automatically export logs to an external destination.



Cloudflare

cf-tme.com

Global Settings

Overview

Analytics & Logs

Traffic

Security

Performance

Edge Reachability

DNS

Workers

Logs

Instant Logs

Version Management

DNS

Email

Spectrum

SSL/TLS

Security

Access

Caching

Traffic

Apps

Zaraz

Web3 New

Collapse sidebar

Chat

← Back to Logpush

Create Logpush Job

✓ Select Data Set — ✓ Select Data Fields — **3 Select Destination** — 4 Enter Destination Info — 5 Prove Ownership

Select a destination

Choose the service that you want to use to receive logs for your domain.

Storage providers

R2 Object Storage Select

Amazon S3 Select

Google Cloud Storage Select

Microsoft Azure Select

S3-Compatible Select

Analytics partners

Sumo Logic Select

Datadog Select

Splunk Select

New Relic Select

Figure 8: Cloudflare Logpush for exporting logs to external destinations

When selecting the vendors for a multi-vendor solution you should ensure you select vendors where the below criteria is met:

- The vendor provides for operational simplicity with a single consistent UI for all operations where users can easily manage and get things done in one place.
- The vendor has useful security analytics to give an understanding of a sites' traffic, security insights, and useful data for troubleshooting.
- The vendor has the ability to export logs/request data to third party clouds/applications.
- The vendor has an API first approach and provides APIs for all operations so tasks can be easily automated.
- The vendor is reputable and can provide effective support and help when needed.
- Employees are trained and have expertise or are comfortable using the vendor's products.

Common deployments

Multi-vendor active-active security and different provider for DNS

The below diagram describes a typical multi-vendor setup in which both vendors are 'active' meaning they are both serving traffic for the same resource (www.example.com) and traffic is split between the two.

On the routing front, this example shows the authoritative DNS living outside of the two providers and load balancing between them. This DNS provider could be self hosted or live on another third party provider. Traffic is directed to each provider by responding to queries for www.example.com with a provider specific CNAME record or static IP for apex domain traffic. To achieve this traffic split, the third party DNS provider does need to have some ability to load balance the traffic. Most major DNS providers will have some mechanism to perform DNS based load balancing with varying degrees of complexity and configurability. This could mean round robin between records in the simplest case, or varying the response based on client location, health check data and more.

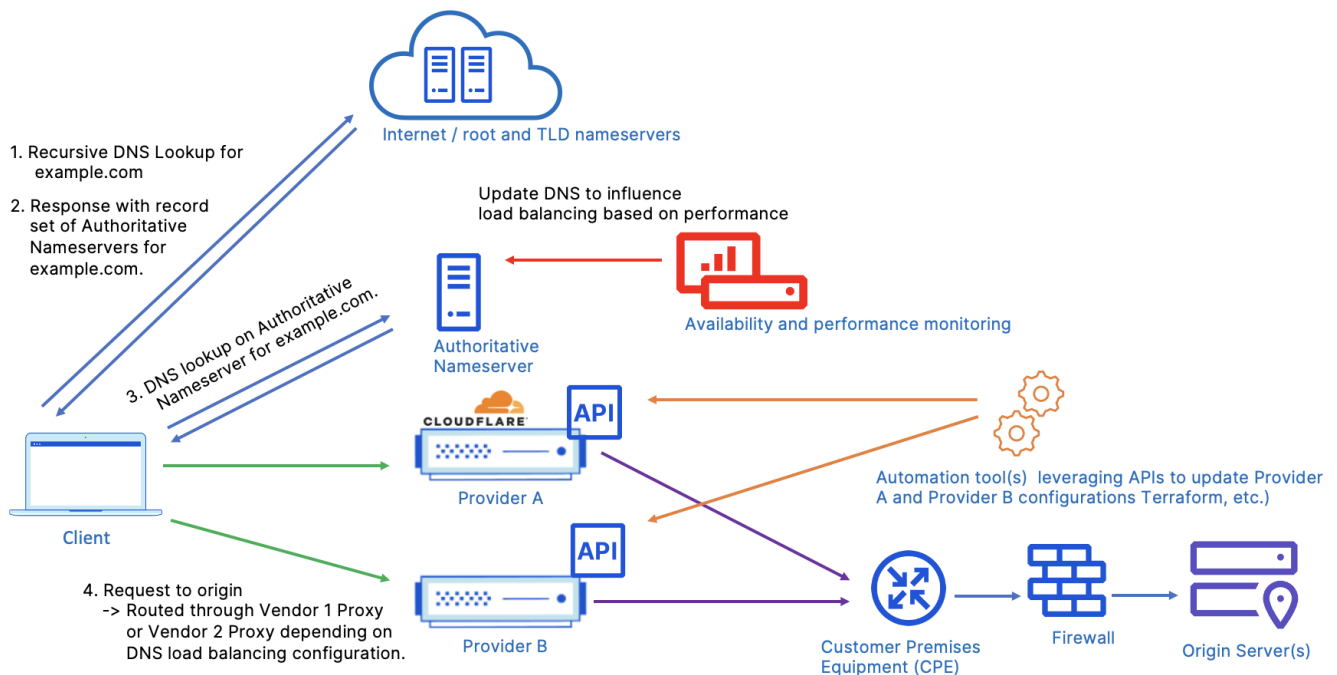


Figure 9: Multi-vendor setup with Cloudflare and another vendor and different provider for DNS

Depending on the authoritative DNS provider, traffic can be evenly split between the two or adjusted dynamically. Oftentimes customers will choose to inform the DNS routing with performance/availability data sourced from a third party monitoring service such as Thousandeyes or Catchpoint and adjust DNS responses based on that data. Third party monitoring services are often used to capture full HTTP request/response metrics to route based

on real-time performance. Traffic can easily be shifted away from a provider by updating the authoritative DNS and waiting for the record TTL to expire.

It's important to note here that the third party services are looking at end-to-end application performance metrics, not just DNS response time or limited data used by DNS resolvers. The DNS records will be updated based on the performance data to reflect the correct security vendor's proxy to point to.

Both providers' configurations are kept in sync by the administrators, pushing out changes via Terraform which makes calls to each provider's API. Keep in mind that while Cloudflare does have full API support for every feature, this may not be the case for every provider.

If only one external DNS provider is used, it does create a single point of failure if that DNS provider has an outage. A way to mitigate this risk is to implement a multi-vendor DNS solution; this is discussed in more detail in the 'Multi-vendor DNS options' section in this document.

Another challenge of a parallel approach is keeping configurations in sync across providers to deliver a consistent end user experience. This means the administrators need to be familiar with the configuration management of both vendors and understand how feature parity can be achieved.

Once traffic is routed to the security and performance service provider via DNS, all security and performance services and respective policies are applied, and the traffic is then routed over the Internet back to the origin where the customer's firewall is allowing IPs specified by each provider.

Multi-vendor active-active security with multi-vendor DNS from same providers

The below example describes a setup where the DNS providers are also the security proxy vendors, and DNS records are kept in sync via zone transfers. A multi-vendor DNS solution is recommended as the preferred and most resilient solution. There are different setups possible between the different DNS vendors and these are discussed in more detail in the 'Multi-vendor DNS setup' section of this document with advantages/disadvantages of each.

In this example, there are multiple authoritative DNS providers used where one is primary and the other is secondary. Per the use of secondary DNS and respective standard, zone transfers easily allow DNS configurations between different providers to remain synced.

In order to point requests to both providers (for the same hosts) in this model, the vendor set up as secondary must be able to overwrite records intended to go through a proxy. Without the

ability to overwrite records as a secondary, the destination for all primary records would remain static and reduce the flexibility and resilience of the overall setup; Cloudflare provides this capability with [Secondary DNS override](#). For example, if the provider such as Cloudflare is set up as a secondary, Cloudflare will have DNS automatically synced to them from the primary via zone transfer, and can use Secondary DNS override to update the A record to point to its own proxy/services.

While DNS based load balancing isn't required here, it's helpful to have at each provider so requests can be predictably split across multiple vendors, otherwise the traffic split is largely dictated by the client resolver nameserver selection.

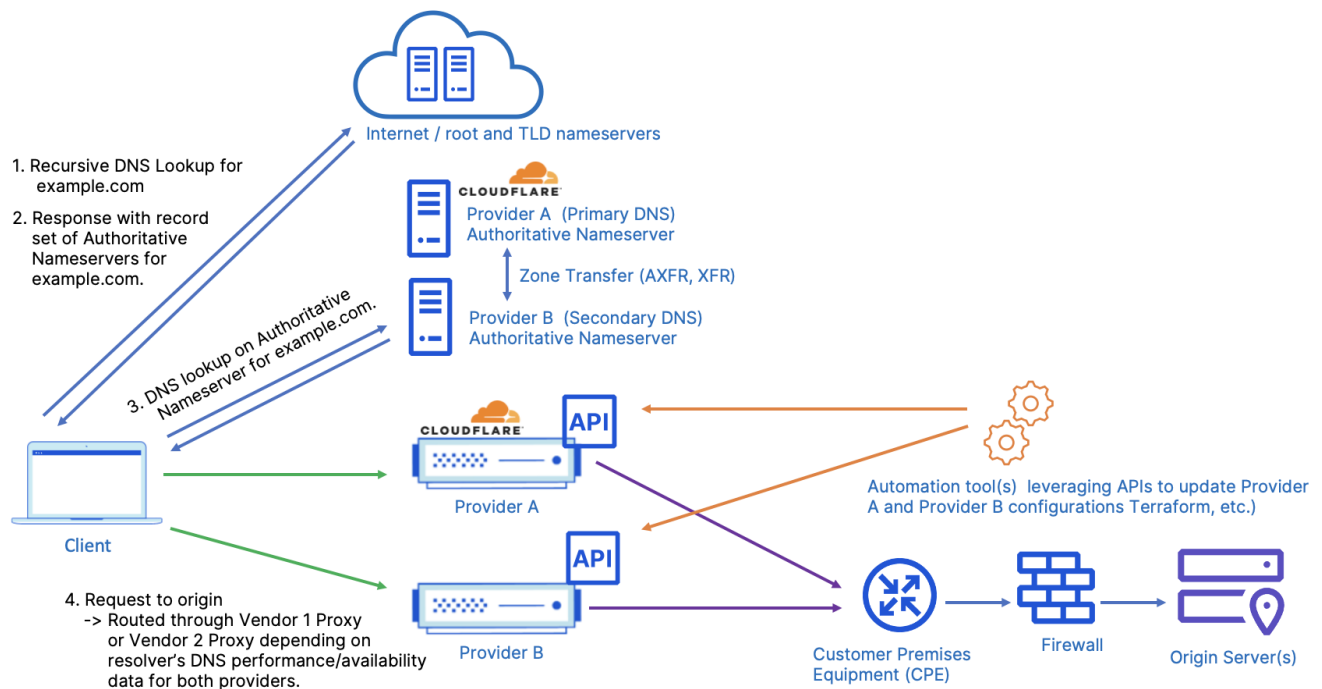


Figure 10: Multi-vendor setup with Cloudflare and another vendor with multi-vendor DNS from same providers

At the authoritative DNS provider, each vendor has their NS records listed and the client will select a nameserver based on their resolver. The resolver will receive the full set of authoritative nameservers upon request. The logic used by most resolvers typically takes into account resolution time as well as availability. In this scenario, the resolvers are used to make the decision on which name server to use based on performance/availability data they already have.

It's important to note here that typically the DNS resolvers have already seen queries and responses associated with the nameservers used. For example, the nameserver the vendor assigns to the customer may already be used by other sites for their authoritative DNS and the resolvers already have a strong historical baseline of performance data to start leveraging immediately.

In this example, we are also seeing records being kept in sync via periodic zone transfers. Cloudflare is able to support both outgoing and incoming zone transfers. Traffic is directed to each proxy by either a provider specific CNAME record or static IP.

The configuration on the DNS side can vary; the different options are discussed in more detail in the next section. DNS can be set up with one provider acting as primary and the other acting as secondary. The DNS provider acting as primary is where all the DNS configuration is done and the secondary DNS receives the configuration copy via zone transfer.

Some DNS providers like [Cloudflare](#) offer the capability where secondary DNS can overwrite the A and AAAA records. This allows the provider to rewrite the A/AAAA record to proxy traffic through a different vendor as desired. In this case, the secondary DNS provider will provide a different response than the primary for the same hostname. This means that depending on what nameserver a client resolver queries, the request will be routed to the vendor's respective network. This allows for flexibility and reduced complexity by relying on the client resolver for traffic steering and failover if the nameservers are slow or unreachable. This comes at the cost of direct control and predictability over what provider a client selects.

Another variation is to have specific applications/hostnames hosted through specific providers. That could mean, in the above example, both the primary and secondary DNS servers have [www.example.com](#) mapped to a Cloudflare address, regardless of which provider resolves the initial DNS query.

Multi-vendor DNS setup options

The important routing decision is dictated by DNS. As discussed, there are multiple configurations possible for a multi-DNS setup. The below assumes you are using two DNS providers which are also the providers for the security solution.

1. Two authoritative - one primary and one secondary

This setup involves setting one provider as a primary and the second provider as a secondary. The purpose of secondary DNS is to support multi-DNS solutions where synchronization between the configurations of primary and secondary is automated.

In this setup both DNS providers are authoritative but only one is primary and the source of truth and where DNS configuration changes/updates are made. The configuration changes/updates on primary are synced to the secondary DNS provider via zone

transfers managed by the provider. DNS of both providers answer DNS queries.

The advantage and main use case with this deployment model is that it uses a standard for synching DNS across multiple providers and was created for just this reason, and the DNS provider is responsible for the zone transfers. This option provides simplicity in maintaining DNS synchronization between providers.

Sometimes customers may decide to use another option due to the following:

- The requirement of updating DNS records when the record management and zone transfer pipeline is down.
- Not wanting to rely on a third party/vendor for the DNS synchronization and desiring more control.
- Having specific restrictions/regulations excluding this option.

This setup is recommended for customers who desire simplicity offered by a secondary DNS and provider for maintaining synchronization.

Pros:

- Uses standard (AXFR, IXFR) to keep DNS synced and done automatically via Zone Transfers.
- Simplicity as the DNS provider is responsible for DNS synchronization.

Cons:

- If the record management and zone transfer pipeline is down, DNS records cannot be updated.
- Some customers do not want to rely on a vendor/3rd party for DNS sync and desire more control and flexibility.

2. Two authoritative - both primary

Some customers may also want to have the added assurance of being able to update DNS records when the record management and zone transfer pipeline is down. They also may not want to rely on a third party/vendor for DNS synchronization and desire more control. In this case, both DNS providers can be used as primary.

In this setup each DNS provider is authoritative and primary. There is no secondary DNS and changes/updates to DNS can be made at either provider; also, both DNS providers answer DNS queries.

Synchronization of the DNS configuration between providers is critical, and in this setup it now becomes the customer's responsibility to keep DNS in sync at both providers. Customers typically do this synchronization with automation tools like OctoDNS,

Terraform, or via custom automation leveraging the vendors' APIs.

This setup is recommended for customers who desire the most flexible and resilient option that supports updating DNS records even when the record management and zone transfer pipeline is down and/or customers who want more control over DNS synchronization.

Pros:

- If control plane is down on one provider, DNS records can still be updated at the other.
- More control and no reliance on DNS provider for DNS synchronization.

Cons:

- More complexity in keeping DNS between providers synced.
- Customer is responsible for DNS synchronization which can be done via automation tools, automated via vendor APIs, or manually.

3. Multiple authoritative - hidden primary and multiple secondary

In a hidden primary setup, users establish an unlisted primary server to store all zone files and changes, then enable one or more secondary servers to receive and resolve queries. Although most of the time the primary is authoritative, it doesn't have to be. In this option, the primary is not listed with the registrar. The primary does not respond to queries and its main purpose is being the single source of truth.

Although the secondary servers essentially fulfill the function of a primary server, the hidden setup allows users to hide their origin IP and shield it from attacks. Additionally, the primary can be taken offline for maintenance without causing DNS service to be disrupted.

This setup is recommended for customers who desire simplicity offered by a secondary DNS and provider for maintaining synchronization. This solution also provides for flexibility in taking the primary offline as needed with less impact.

Pros:

- Allows customers to maintain DNS record management on their infrastructure and use standard to keep DNS synced automatically via Zone Transfers.
- Primary is used only for source of truth and maintaining DNS records and can be taken offline for maintenance /administration.

Cons:

- If the record management and zone transfer pipeline is down, DNS records cannot be updated.
- Some customers do not want to rely on a vendor/3rd party for DNS sync and desire more control.

Configuration and management best practices

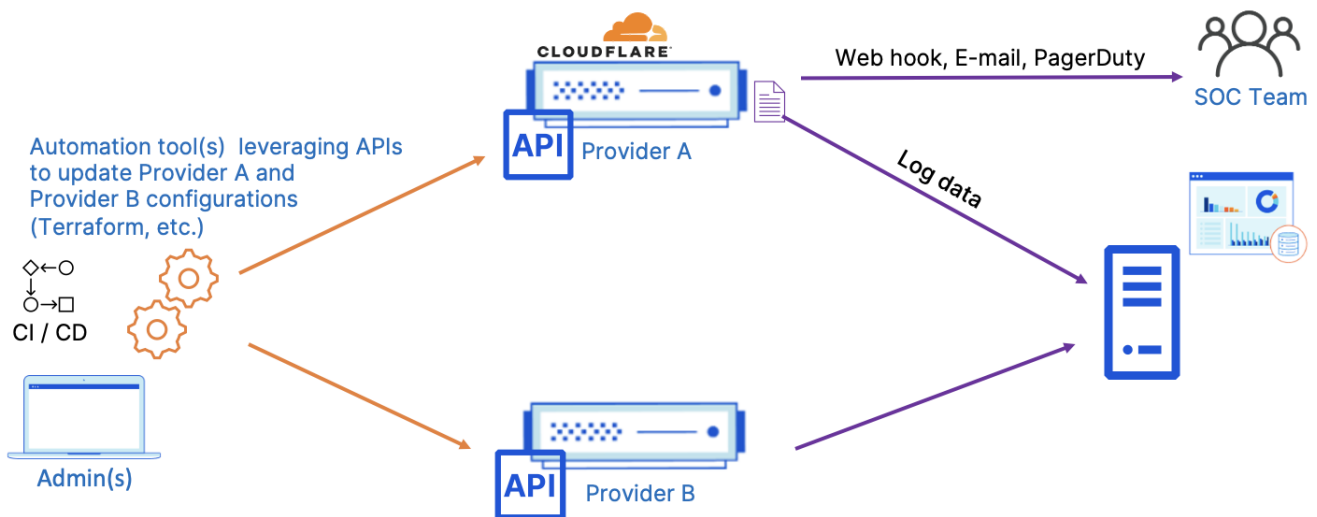


Figure 11: Configuration via Terraform for multi-vendor setup with Cloudflare and other vendor

Figure 11 depicts a typical pattern seen when managing configurations across both Cloudflare and other providers in parallel. In this example, we are assuming that the same workloads are being split through both providers and the admin team is updating both configurations via API through Terraform. This can also be tied into an internal CI/CD pipeline to match your typical developer workflow. All Cloudflare functions can be configured via API and are delivered first via API. This diagram also depicts logs being sent to a common SIEM and native alerting functions that can be delivered via e-mail, webhook, or PagerDuty for alerts based on performance, security or administrative criteria.

With the wide variety of customization options Cloudflare provides (Ruleset Engine, native features, Worker customizations), Cloudflare can likely meet feature parity with most other major vendors out in the market, however it's not guaranteed that these features will be configurable in the same manner. This is where working closely with your Cloudflare account team becomes critical in understanding the key differences in operation and best practices to align your workflow with Cloudflare.

Connectivity options

For a multi-vendor offering it's important to consider the methods that each provider offers for connectivity to the origin(s) and the trade offs in security, performance, and resiliency. Cloudflare offers several options that fit most use cases and can be deployed in parallel with per application (hostname/DNS record) granularity to fit a hybrid customer environment.

Internet (default)

In the most basic scenario, the proxy will simply route the traffic over the Internet to the origin; this is the default setup for all vendors. In this setup the client and origin are both endpoints directly connected to the Internet via their respective ISPs. The request is routed over the Internet from the client to the vendor proxy (via DNS configuration) before the proxy routes the request over the Internet to the customer's origin.

The below diagram describes the default connectivity to origins as requests flow through the Cloudflare network. When a request hits a proxied DNS record and needs to reach the origin, Cloudflare will send traffic from the network over the Internet from a set of Cloudflare owned addresses.

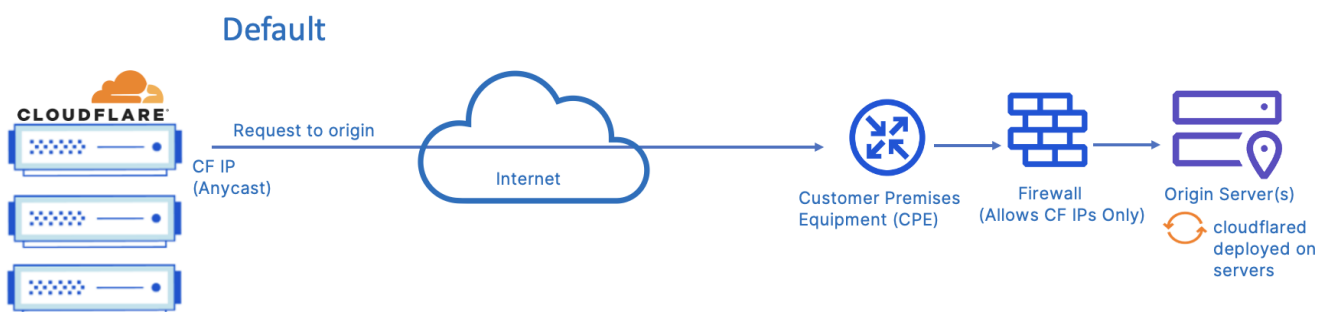


Figure 12: Connectivity from Cloudflare to origin server(s) via Internet

Optionally, customers can also choose to leverage [Cloudflare Aegis](#), which allocates customer-specific IPs that Cloudflare will use to connect back to your origins. We recommend whitelisting traffic from only these networks to avoid direct access. In addition to IP blocking at the origin side firewall, we also strongly recommend additional verification of traffic via either the "Full (Strict)" SSL setting or mTLS auth to ensure all traffic is sourced from requests passing through the customer configured zones.

Cloudflare also supports [Bring Your Own IP \(BYOIP\)](#). When BYOIP is configured, the Cloudflare global network will announce a customer's own IP prefixes and the prefixes can be used with the

respective Cloudflare Layer 7 services.

Private connection - tunnel or VPN

Another option is to have a private tunnel/connection over the Internet for additional security. Some vendors offer private connectivity via tunnels or VPNs which can be encrypted or unencrypted; these vary in complexity/management and require additional security/firewall updates to allow for connectivity. A traditional VPN setup is also limited via a centralized vendor location back to the origin.

Cloudflare offers [Cloudflare Tunnel](#) which is tunneling software that provides an encrypted tunnel between your origin(s) and Cloudflare's network. Also, since Cloudflare leverages Anycast on its global network, the origin(s) will, like clients, connect to the closest Cloudflare data center(s).

When you run a tunnel, a lightweight daemon in your infrastructure, `cloudflared`, establishes four outbound-only connections between the origin server and the Cloudflare network. These four connections are made to four different servers spread across at least two distinct data centers providing robust resiliency. It is possible to install many `cloudflared` instances to increase resilience between your origin servers and the Cloudflare network.

`Cloudflared` creates an encrypted tunnel between your origin web server(s) and Cloudflare's nearest data center(s), all without opening any public inbound ports. This provides for simplicity and speed of implementation as there are no security changes needed on the firewall. This solution also lowers the risk of firewall misconfigurations which could leave your company vulnerable to attacks.

The firewall and security posture is hardened by locking down all origin server ports and protocols via your firewall. Once Cloudflare Tunnel is in place and respective security applied, all requests on HTTP/S ports are dropped, including volumetric DDoS attacks. Data breach attempts, such as snooping of data in transit or brute force login attacks, are blocked entirely.

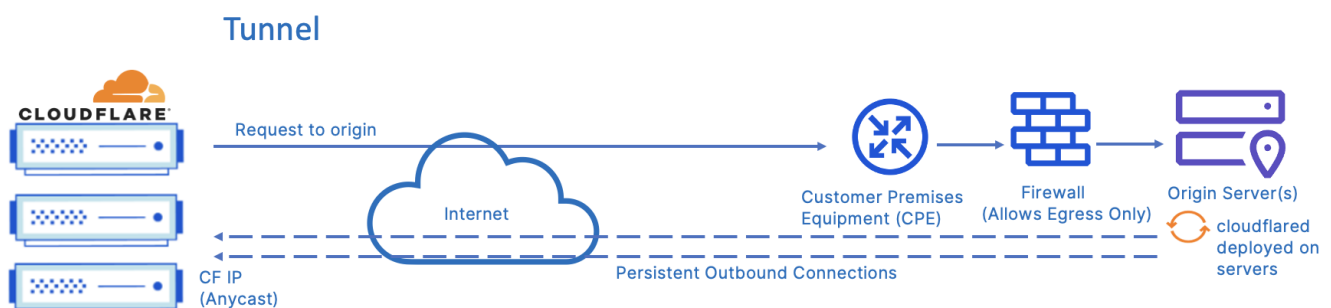


Figure 13: Connectivity from Cloudflare to origin server(s) via Cloudflare Tunnel

The above diagram describes the connectivity model through Cloudflare Tunnel. Note, this option provides you with a secure way to connect your resources to Cloudflare without a publicly routable IP address. Cloudflare Tunnel can connect HTTP web servers, SSH servers, remote desktops, and other protocols safely to Cloudflare.

Direct connection

Most vendors also provide an option of directly connecting to their network. Direct connections provide security, reliability, and performance benefits over using the public Internet. These direct connections are done at peering facilities, Internet Exchanges (IXs) where Internet Service Providers (ISPs) and Internet networks can interconnect with each other, or through vendor partners.

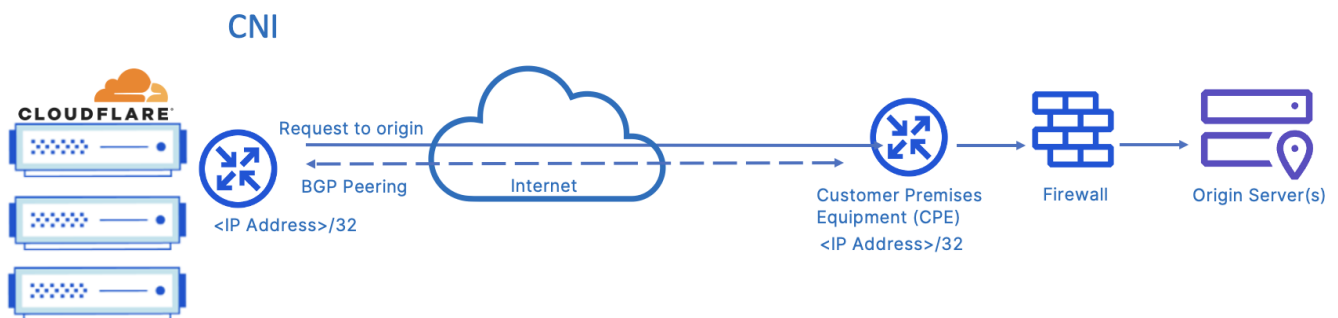


Figure 14: Connectivity from Cloudflare to origin server(s) via Cloudflare Network Interconnect (CNI)

The above diagram describes origin connectivity through [Cloudflare Network Interconnect \(CNI\)](#) which allows you to connect your network infrastructure directly with Cloudflare and communicate only over those direct links. CNI allows customers to interconnect branch and headquarter locations directly with Cloudflare. Customers can interconnect with Cloudflare in one of three ways: over a private network interconnect (PNI) available at [Cloudflare peering facilities](#), via an IX at any of the [many global exchanges Cloudflare participates in](#), or through one of our [interconnection platform partners](#).

Cloudflare's global network allows for ease of connecting to the network regardless of where your infrastructure and employees are.

Additional routing and security options

Most vendors also provide additional capabilities for enhanced/optimized routing and additional security capabilities when communicating with the origin. You should check with respective vendor documentation to confirm support if parity is expected in terms of performance and security capabilities.

Cloudflare offers [Argo Smart Routing](#) for finding and using optimized routes across the Cloudflare network to deliver responses to users more quickly and Authenticated Origin Pulls (mTLS) to ensure requests to your origin server come from the Cloudflare network

Argo Smart Routing

Argo Smart Routing is a service that finds optimized routes across the Cloudflare network to deliver responses to users more quickly.

Argo Smart Routing accelerates traffic by taking into account real-time data and network intelligence from routing over 28 million HTTP requests per second; it ensures the fastest and most reliable network paths are traversed over the Cloudflare network to the origin server. On average, Argo Smart Routing accounts for 30% faster performance on web assets.

In addition, Cloudflare CDN leverages Argo Smart Routing to determine the best upper tier data centers for Argo Tiered Cache. Argo Smart Routing can be enabled to ensure the fastest paths over the Cloudflare network are taken between upper tier data centers and origin servers at all times. Without Argo Smart Routing, communication between upper tier data centers to origin servers are still intelligently routed around problems on the Internet to ensure origin reachability. For more information on Argo Smart Routing as it relates to CDN, see the [Cloudflare CDN Reference Architecture](#).

Authenticated Origin Pulls (mTLS)

Authenticated Origin Pulls helps ensure requests to your origin server come from the Cloudflare network, which provides an additional layer of security on top of [Full](#) or [Full \(strict\)](#) SSL/TLS encryption modes Cloudflare offers.

This authentication becomes particularly important with the [Cloudflare Web Application Firewall \(WAF\)](#). Together with the WAF, you can make sure that all traffic is evaluated before receiving a response from your origin server.

If you want your domain to be [FIPS](#) compliant, you must upload your own certificate. This option is available for both [zone-level](#) and [per-hostname](#) authenticated origin pulls.

Summary

To summarize, a successful multi-vendor strategy for application security and performance requires careful consideration of your business objectives, infrastructure requirements, and vendor capabilities. There are several options to choose from when deploying a multi-vendor strategy with various advantages and limitations to each. Cloudflare can support these configurations by delivering services through the Cloudflare Global Network that are highly resilient, performant, and cost effective to fit your organizations multi-vendor strategy.