Whitepaper

CDN buyer's guide

Many CDNs have similar features. Here are the questions that will help you choose the best, and ignore the rest.

fastly®

Introduction

It can be difficult to navigate the different options when you're looking to improve your CDN. Many CDNs offer the same types of services, but they're not all the same. Many customers who select the wrong CDN end up facing the following challenges:

- Slow site, application, or streaming performance
- Too many technical issues that negatively impact uptime and end-user experience
- Lack of self-service to fix problems or make changes to unlock new features and innovation
- Elevated costs associated with with legacy solutions

Use this checklist to select the best option that can meet your needs today, and be ready to grow with you in the future (without hitting you with surprise limitations or charges!). Any CDN can cache an image for you. Here is a list that will help you separate the best from the rest.

Don't forget to keep reading below the list for better explanations of each of these items, and some links to helpful resources if you want to learn more about them!

CDN buyer's guide checklist

Infrastructure, network, and platform

- ☐ Modern architecture and network design
- ☐ Fully software defined infrastructure
- ☐ Operates on a single global network
- ☐ Unified platform customer experience with everything under one control panel
- ☐ Uses the most modern protocols for speed and privacy such as HTTP3, QUIC, TLS 1.3, and proactively adopts when new options arrive

Read the details →

Performance

- ☐ High-capacity Points of Presence (POPs) with larger caches
- ☐ POPs that use solid state storage drives (SSDs) for faster content delivery
- ☐ Ability to cache static, dynamic and event-driven content
- ☐ Static and dynamic content compression
- ☐ Rapid image optimization at the edge
- ☐ Optimized load balancing

Read the details →

Origin offload and shielding

- ☐ Elect a specific POP on the network as your origin shield
- ☐ Request collapsing to further reduce calls to your origin servers
- ☐ Content is in cache longer without being deleted as least recently used (LRU)
- ☐ Multi-CDN shielding compatibility to maximize offload even in multi-CDN situations

Read the details →

Instant and real-time

- ☐ Websockets support and acceleration that works with your WebSockets infrastructure
- ☐ Websockets support and acceleration managed over HTTP without requiring your own WebSockets infrastructure
- ☐ Purge content instantly on par with a 150ms mean purge time to clear cache globally

Read the details →

Control, configurability, and developer friendliness

- ☐ Self-service control and configuration
- ☐ API friendly
- ☐ CI/CD friendly and works with the tools you already use, with no hidden charges
- ☐ Terraform support

Read the details →

CDN buyer's guide checklist

Reliability, availability, and resilience

- ☐ Ongoing Business Continuity Planning (BCP) testing programs for network resilience
- ☐ Control and data plane failovers
- ☐ Data center failovers
- ☐ Power grid interruption failovers
- ☐ Automated network re-routing and auto-healing
- ☐ Automated and immediate network-level DDoS protection

Read the details →

Real-time visibility, logging, and insights

- ☐ Instant log streaming (not limited by time lag or batch reporting)
- ☐ Custom self-service reporting, data visualizations, and performance alerts
- ☐ Multi-CDN traffic management and cloud infrastructure analytics
- ☐ Simple integrations into the logging endpoints and services you already use
- ☐ Edge compute metrics
- ☐ Origin and domain level metrics
- ☐ Complete performance visibility from origin to edge

Read the details →

Pricing

- ☐ No overages, no surprises, no fine print
- ☐ Straightforward pricing and packaging available
- ☐ Broad feature set that makes vendor consolidation possible
- ☐ No surprise professional services costs
- ☐ Available to purchase through marketplaces like AWS and GCP

Read the details →

Customer success and support

- ☐ High customer satisfaction
- ☐ Fast and effective customer support
- ☐ Customer support that doesn't push for surprise professional services fees
- ☐ Comprehensive onboarding and migration support
- ☐ Extensive documentation that is kept up to date
- ☐ Active customer communities

Read the details →

Learn more about why it's important

Infrastructure, network, and platform

The performance and feature set a CDN can offer has hard limitations based on the technology, architecture, and software of the network it runs on. This can be important for its performance right now, but even more important if it will impede its ability to grow with you and continue to offer the services you need in the future. The internet continues to evolve quickly, and user expectations for the performance, safety, and reliability of their internet and streaming experiences grow as well. The cost of switching vendors is significant, so it's important to choose one that will meet your needs for the long term.

Here are some ways you can ask about the modernity, limitations, and technical debt of a CDN network. The architecture and network design should be modern, using technologies that allow the network to run on fewer, more powerful, and more efficient points of presence (POPs) rather than running many underpowered POPs inefficiently. The network infrastructure should be fully software defined so that no additional latency is introduced by hardware components like load balancers or routers that can't be fully controlled or optimized. This is also critical for future growth because a software-defined network means that every single function can be updated, upgraded, and optimized as the needs of the customers and the internet change over time.

Operating all functions and products on a single global network is another important sign because of the efficiency of only having to maintain and evolve one global network. If some features sit on different networks, then their costs for maintaining and updating their networks is divided, diluted, and more expensive. Beyond that, hopping between networks introduces more latency. Performance may be slower, innovation will be slower, and the costs from all of this inefficiency will be passed on to the customers.

Lastly, the platform should handle many things on behalf of the customer, like adopting the most advanced, efficient, and secure protocols. Right now this includes examples like HTTP3, QUIC, and TLS 1.3, but there should also be a guarantee to proactively migrate to the newest, fastest, and safest options to continually provide better service without requiring effort from customers.

Performance

There are three huge benefits to serving as much content as quickly as possible from your CDN's cache instead of from your own servers.

1. **You save potentially huge amounts of money on egress charges and hardware costs if you shift more to your CDN.**
2. **Faster performance helps you grow your revenue by reducing customer abandonment and improving conversion rates.**
3. **With the help of a CDN you can scale instantly, on-demand when required without having to build or maintain a huge infrastructure for peak traffic events.**

Many CDNs are limited in terms of how much they can serve from cache, and how quickly they can serve it. It starts with how their network is built, down to the architecture and components selected. Building a global network is expensive, and many CDNs cut corners, or were built a long time ago, or simply lacked the foresight to use cutting edge, high-capacity components. It's a bigger investment when building the CDN network, but having more storage for content on each POP helps CDNs keep content in cache longer and reduce calls to your origin. SSDs mean that the POPs can read content out of memory significantly faster than if they used cheaper spinning disks. More powerful computing allows POPs to do more, and do it incredibly fast. This includes fundamental things like automatically optimized load balancing and the ability to cache dynamic and event driven content (not just static content). But it also means handling image processing at the edge, or performing static and dynamic content compression at the edge to shrink payloads without introducing latency.

Along with the performance benefits you can also reduce your operating costs, including capital expenditures. Outsourcing your ability to respond to peak traffic events to your CDN means that you don't have to build out your infrastructure at origin to handle extreme traffic peaks and watch most of your capacity sit idle and burn through your budget most of the time while operating normally.

Origin offload

One of the key benefits of a CDN is to serve as much as possible from cache, and offload as much traffic as possible from your servers at origin. This can translate into cost savings through (sometimes massive) reductions in egress charges, but it can also reduce your costs in other ways. If your origin servers are protected from traffic spikes, bot attacks, and generally made to handle a small percentage of requests, then you can reduce the number of servers you're running and cut your CapEx and maintenance expenditures as well.

Choosing one specific POP to serve as an origin shield gives you a guarantee that all other requests, even from other POPs on the CDN network, will go to that POP rather than your origin server in order to provide the maximum amount of offload from your servers, and ensure that the CDN is handling as much work for you as possible. Another feature you'll want that helps with offload is request collapsing. This bundles multiple requests for the same piece of content across the entire network so that your origin only has to serve that content once rather than individually for each request. This is particularly useful if a piece of content expires out of cache and then at a later point there are many concurrent requests for it across the network.

You can also look for CDNs that can keep your content live and in cache longer. "Least recently used" (LRU) refers to content that a POP deletes from its memory because it's in low demand, in order to make room for other content that's in higher demand. More powerful POPs are built with higher end and higher capacity components that can store more, and therefore expire it less often. With more powerful POPs, your content is subject to deletion as LRU less often. The CDN will store it longer and handle more requests without going back to your servers. By contrast, POPs with less storage capacity must delete LRU content much more often to make room for the other content they're serving. Every time your content cycles out of their cache they need to go get it from your origin servers. If hundreds and hundreds of POPs on a global network are regularly needing to re-request content from your origin because it's always cycling out of cache, it will add up to significant costs.

Many organizations need to use multiple CDNs in combination. Some CDNs can be difficult to manage efficiently alongside their counterparts or don't have features to help you coordinate a multi-CDN strategy, but that's not true for all CDNs, especially with regard to origin shielding. Selecting a CDN that helps coordinate the shielding between your origin and ALL of the CDNs that are a part of your strategy is key to maximizing both your performance and cost efficiency.

Instant and real-time

In the past you would pay a CDN to distribute your content a little faster and cheaper than you could on your own, but "faster" isn't good enough anymore. Organizations are demanding real-time distribution and control of their content and communications. The term "real-time" gets thrown around a lot, but in this case it doesn't mean faster – it means instant and with instant global scalability. Some examples of one-to-many push communications are online gaming, stock prices and other financial tickers, sports scores, in-app chat experiences, real-time location updates, and more. It also means being able to purge data that is out of date instantly and globally.

It used to be good enough to be quick, but today CDNs need to be able to perform instantly. This includes data and content services that instantly push data out globally, and the ability to purge content globally in an instant. Whether you use WebSockets and want better control and performance. Or maybe you want the option to have all of the same features and benefits without running your own WebSockets stack; keep your options open with a CDN that can deliver all the same value over HTTP without requiring a WebSockets stack at origin.

If you already use a separate vendor for stateful, instant communication, this is also an opportunity to get two improvements: a performance upgrade and some vendor consolidation. Point solutions for instant communications have smaller, less powerful networks for distributions than a best-in-class CDN. You can save costs, simplify your budget, and consolidate your customer support into one place while moving your real-time comms to a more powerful network.

Supporting truly instant content isn't just about distributing the content, it's also about purging it. If you can get it out there instantly, you also need to be able to pull it down instantly and globally to ensure accuracy and prevent your users from having confusing or unfortunate experiences with out-of-date content.

Control, configurability, and developer friendliness

Evaluating CDN options isn't just about the features available for content delivery, it's also about the features available for your team as they interact with the CDN. You want to make sure that your selection helps you reduce the number of manual and labor-intensive processes. If your team can spend less time and allocate fewer people to managing your CDN, then you can reallocate resources and accomplish more of your other priorities. You should also look at how much can be done with self-service capabilities rather than making requests to your vendor for a change. This matters for saving time and implementing changes faster, but it can also protect you from racking up unexpected professional services charges to make configuration changes that you could do on your own with a better CDN.

Having well built and documented APIs is one big way that a CDN can give you easy access, and work with the existing continuous integration and continuous delivery (CI/CD) workflows your team is already using. For example, if you use Terraform, is that supported? Does it work with the logging endpoints you already use? Can you easily pipe your data to wherever you want it?

Reliability, availability, and resilience

Reliability seems like it should table stakes, but not all CDNs are equally prepared and ready to respond to preserve performance in the face of problems. Make sure to select an option that has backup plans and failovers and even more backup plans so that your organization and users are protected. You want to select a vendor who has an active Business Continuity Planning (BCP) practice, and is transparent about their planning. BCP is proactive resilience planning in order to still be able to perform at expected levels, even during a disruptive or catastrophic incident. This planning should include things like resiliency, redundancy, and failover planning for the control plane, the data plane, data centers, and in case of power grid interruptions at any of those points.

The CDN network itself should also have instant, automated responsiveness to route around problems like internet outages, traffic anomalies, latency issues, and other forms of "internet weather," so that network problems don't impact your performance and availability. The network should also be able to absorb and mitigate DDoS attacks automatically without requiring human intervention, which can add a critical delay into the response process.

Visibility, logging, and insights

It's not enough for data to exist – it needs to be high quality, and accessible so that you can derive valuable insights. Some of the most important things to check are around real-time access and precision. Avoid situations where data is only available at a lag – even seconds can hamper your decision-making, and minutes are unacceptable in critical decision-making moments. Even worse is if it's only accessible through delayed batch reports. Any amount of latency can create or prolong web and application downtime when you're trying to diagnose a problem and troubleshoot quickly. If you have to make professional services requests, and pay them to deliver your data, then you're paying extra for a solution that is already unacceptable.

You should have access to granular views of all data – both real-time and historic. You should have access to full visibility from origin to end-user, including domain-level metrics, so that you can understand where your egress charges are coming from, and lower them. You should also get insight into what's happening within multi-CDN environments without jumping between dashboards every time you have a question. The data should be customizable to fit your needs, and be available however you need – supporting multiple protocols and ready-made integrations with the logging endpoints and services you already use. There should also be pre-built dashboards provided that deliver value right away and customizable dashboards to streamline workflows.

If your CDN offers other services like security and computing solutions, your data availability should extend just as well into the security logging and computing activity with just as much control and ability to draw insights.

Pricing

Pricing should be clear, with no hidden charges, and no overages or surprises after the fact. Many vendors say there are no overages, but in the fine print you still find cases where you may encounter huge surprise bills for events that may be out of your control. Look for a vendor that can protect you from all surprise overages without exception, and who is dedicated to helping you find the right plan for the service you need. Customer satisfaction can be a good indicator of this, so check out the next section as well.

Professional services can be useful for some projects you want to take on with help from your vendor, but it shouldn't be something that is required for simple things that a better CDN would let you update on your own, or handle through a better customer support offering.

If you already spend a lot of your budget with AWS or GCP you should look for a CDN that you can purchase through their marketplaces. This can simplify your billing and allow you to use budget or credits that are already locked into those systems rather than freeing other dollars from the organization. Sometimes you can even find discounts or deals from the marketplace to incentivize you to spend your budget there.

Customer success and support

Customer success, support, and satisfaction are a good way to investigate whether all of the other claims a CDN vendor makes can be trusted. You want to check if they have a high level of customer satisfaction overall, and how their customer support is viewed. Do their customers think their support is fast, helpful, and comprehensive? Do they feel like they have a working partnership?

You should be able to have a clear understanding from the start about what will fall under professional services engagements and what will not, and you want to see a wealth of high quality and up-to-date resources for customer onboarding, migration, and product documentation. You should also look for an active customer community where you can ask questions to other customers along with experts from the vendor, and gain insights from what your peers are doing.

Get in touch to learn more

Additional resources

Learn more about a few of the topics mentioned in the buyer guide:

1 Doing real-time stateful communications at the edge, with or without WebSockets – read the blog post

2 Image Optimization at the edge – read the ebook

3 Fastly’s approach to preventing outages with resilient architecture – read the blog post

4 A guide to the modern CDN – get the ebook

5 Total economic impact: see how choosing the right CDN can deliver 189% ROI – get the report