VARNISH SOFTWARE

White paper:

HOW TO BUILD YOUR OWN CDN

– With Varnish Private CDN

How to build your own CDN – with Varnish Private CDN

Introduction

With Varnish® Software's DevOps-friendly Private CDN solution you can build your own content delivery network and take back control of your content, maximize performance and manage costs

Content delivery has followed a steady evolution over time. Its trajectory began with content delivery through external providers' networks (CDNs), moving toward a multi-CDN approach. Now it has evolved to include a mix of content delivery solutions: more do-it-yourself, private solutions and hybrid content delivery strategies. Content delivery, and the methods powering it, have been guided by user demand and flexibility, allowing companies the ability to "go it alone" or pick and choose CDN elements based on needs of their unique audiences and requirements.

To meet the potential inherent in these demands, Varnish™ Software's Varnish Private CDN™ solution combines high-performance caching nodes with edge-computing logic. As content delivery challenges and demands change rapidly, Varnish Private CDN is designed to work flexibly because there is no "one-size-fits-all" content delivery solution any more. Whether meeting the needs of a hybrid CDN strategy, or a fully private CDN, Varnish Private CDN is a DevOps-friendly option that meets the unique content delivery needs of any content provider.

What you will learn:

- The evolution of the content delivery network
- The changing content delivery landscape
- Why you would decide to build your own CDN solution
- What you need to build your own hybrid or private CDN

Figure 1: The evolution of CDNs has followed a steady trajectory over time where the private CDN has now surfaced as a feasible option for companies wanting better control of their own content and the content delivery costs.

Why take your CDN in-house?

Enabling flexibility in content delivery is increasingly important to delivering a world-class user experience to consumers. Some of the challenges that drive the decision to go in-house include:

- **Availability –** For content providers that require guaranteed availability of all content to all users at all times, taking control of delivery is critical. While there may be some comfort in dressing down a CDN that suffers a major outage, the reality is that users blame the service they are trying to use, not the network provider. CDNs tend to locate their PoPs in the same places (i.e. large cities in major countries); as a result, during high traffic periods, they share the same peering relationships, which leads to congestion. Selecting your ISP, you need to ensure that you've made a selection that focuses on stable, always-available performance, as this will form the backbone of your business.

- **Performance –** There are four performance-oriented areas to consider:

  1. **Peering congestion** negatively impacts performance from the user's perspective. Creating your own distributed network, reaching the more far-flung audience centres, increases your ability to deliver when the CDNs are overwhelmed.

  2. **Closeness to your most important users** is critical to serve your content from the closest location possible to the main concentrations of content consumers: the closer the content is, the more quickly and efficiently it will be delivered to your audience, i.e. the closer the content is to consumers, the smaller the network latency.

  3. **Highly dynamic content does not lend itself well to CDN coverage:** in a shared environment, it is hard to get the caching right. Using a private network with Varnish caching allows you to cache even dynamically generated content, providing not only faster throughput on a generalized basis, but also local caches that can keep the service running in the event of a major origin server outage

  4. **Long-tail content lends itself better to a Private CDN setup:** long-tail content should be delivered at the same performance level as other elements of your service, making a Private CDN a sensible investment. A number of factors in Varnish help with diverse, long-tail content, such a prefetch functionality and flexible VCL to implement strategy at the edge of the architecture.

- **Costs –** Every additional gigabyte of traffic delivered over a CDN has a price: beyond progressive pricing tiers, there is no economy of scale to be had. By contrast, as traffic share moves to a private network, the capital investment gets shared more broadly, and the overall cost to operate is reduced. Many customers find that they can achieve significant cost savings by offloading expensive traffic from commercial CDNs to their private networks. You can quickly do the math on whether building out your own private network makes financial sense. You will need to calculate the cost of buying your hardware or using virtual server hosting services such as AWS EC2, establishing PoPs, and operating your network; then compare it to the price of simply contracting with one or more commercial CDNs. A cost comparison between the two represents your foundational information. It is key to note that as the level of traffic increases, the additional operating expense of contracting with a commercial CDN will increase accordingly. Also, many commercial CDNs have put a very high additional price tag to include security features and support, such as SSL/TLS, despite these being a must-have for modern data transport. With a private network,you will start to see economies of scale and avoid this extra expense.

- **Scale –** At scale companies like Netflix, Apple and Facebook have already built their own solutions rather than using a commercial CDN. This gives them increased control, visibility, and the ability to place their PoPs based on their own unique needs, rather than the generalized needs of a CDN's entire customer base.

- **Security –** Many customers find that having their valuable content on a multi-tenant CDN is not optimal for security and privacy. A private CDN structure can offer various security safeguards to provide valuable protection against DDoS and other attacks as well as more control over the security measures implemented.

- **Flexibility –** With VCL, a Varnish-only feature, you gain the unique flexibility to add tailor-made solutions and features to every layer of your cache, including at the edge, for more fine-grained control over your content and its delivery.

Whether meeting the needs of a hybrid CDN strategy, or a fully private CDN, Varnish Private CDN is a DevOps-friendly option that meets the unique content delivery needs of any content provider.

Why take your CDN in-house? Enabling flexibility in content delivery is increasingly important to delivering a world-class user experience to consumers.

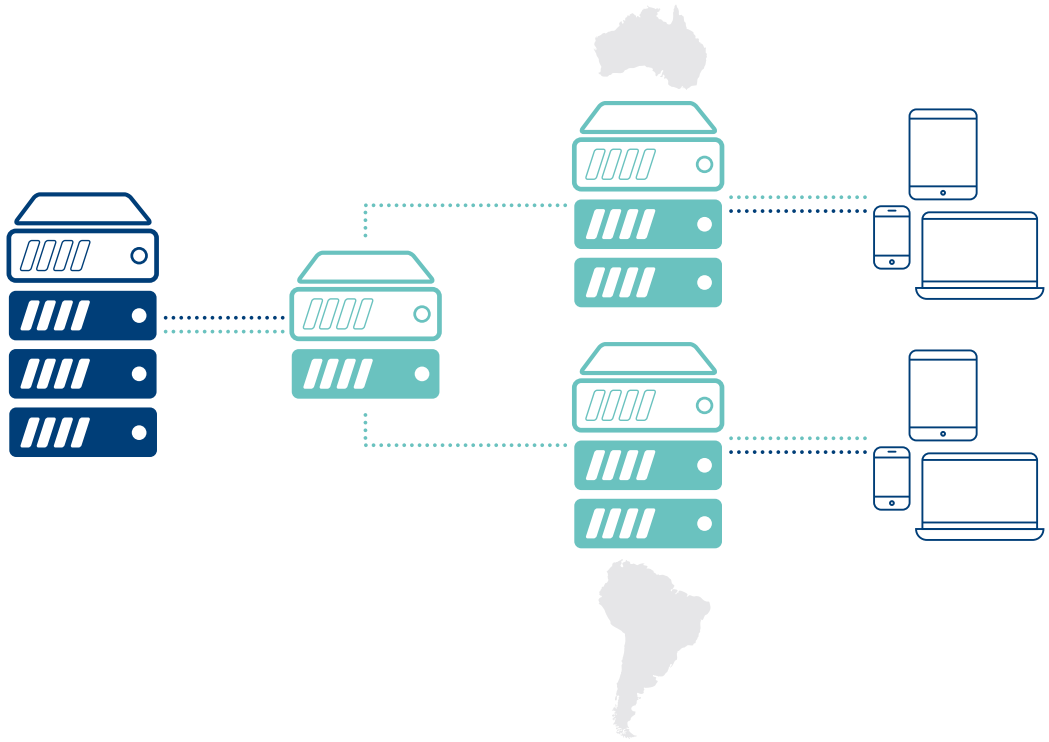

Figure 2: Using a combination of high-performance caching nodes with edge-logic computing and smart load balancing, you can build your own content delivery solution that delivers world-class user experience

In short, companies managing high volumes of internet traffic, who once were best served by contracting with CDNs, are now finding that in the areas of performance, cost, scale, and security, they are better served by doing it themselves. Where once the idea of building and maintaining one's own content delivery network was prohibitive in terms of cost, time, hardware and expertise, it is now affordable, increasingly common and relatively quick to deploy (a private CDN can be built in as little as one afternoon).

Where once the idea of building and maintaining one's own content delivery network was prohibitive in terms of cost, time, hardware and expertise, it is now affordable, increasingly common and quick to deploy.

How to build your own private CDN in five easy steps

Building your own CDN is no longer a complicated and time consuming undertaking. In five easy steps, you can have a private CDN up and running, which can be deployed to augment your commercial CDN or to replace it entirely.

High-performance content delivery will rely on serving content following web performance best practices to decide whether to deploy your CDN in the cloud, on-premise or a hybrid approach of both.

Five elements are required to build your own CDN:

1. Deciding between a cloud or on-premise solution
2. Choosing ISP and routing
3. Origin server and Points of Presence (PoPs)
4. Private CDN and caching software
5. Configuration

1. Deciding between a cloud or on-premise solution

The first decision you will have to make is whether to deploy your content delivery network using cloud instances or physical machines. This choice will come down to how you answer a series of questions, and you will have to ask yourself this question both for the origin server(s) and then for your PoPs. It's not a mutually exclusive decision; you can decide to implement a portion of your CDN using one (or more) cloud solution(s) while another portion lives on-premise.

You will want to consider the following in making this decision:

- **Availability:** Does the cloud provider offer machines in the locations/regions I want to serve content to? Does my own company have data centre(s) in the region where I want to deliver content?
- **Performance:** Consider what kind of content you plan to deliver and how it will affect performance using cloud instances versus physical machines.
- **Costs:** Is it cheaper to rent the infrastructure or own it? In the short-term and in the long-run? Also consider the cost of maintenance in this equation.
- **Scale:** Which of these approaches best aligns with my strategy for scaling up the business?
- **Security:** What are my security considerations, and what are the pros and cons of each approach?

2. Choosing ISP and routing

In a private CDN setup, you will bring your own ISP and routing solution (or already have this if you are an ISP yourself). This ensures that you will be able to choose the network that best suits your needs and setup.

Some key considerations in selecting an ISP that can also route each request to the closest PoP:

- **Geographic presence:** Find an ISP that has coverage in the region(s) you want to serve
- **Connection speed:** Ensure you have enough speed for both your average day and peak times as well
- **Availability and reliability:** As with all parts of your infrastructure an ISP needs to be always available as it will be the backbone of your business
- **Cost efficiency:** Map out service costs carefully, as costs from an ISP will vary widely depending upon a number of factors, and you will have to find the proper balance between services and costs

3. Origin server and Points of Presence (PoPs)

At its most basic, a CDN is a huge cache with distributed nodes that help you deliver content faster to end users.

The origin server needs to be as close as possible to you to ensure you have full control of your content. PoPs are what enable you extend to your reach and shorten the distance between your content and your end users; to achieve the goal of high performance content delivery, there are four main drivers to this decision:

1. Understanding where your current audience is located. Where is your most important audience and where do you expect to see growth?
2. If needed, projecting locations to which you want to place PoPs.
3. Measuring the network latency between your origin server and your audience.
4. Understanding the potential traffic demands at each location. Perform analysis to optimize your infrastructure.

4. Private CDN and caching software

The Varnish Private CDN solution includes several Varnish components that, as a package, give you what you need to develop your own CDN solution:

1. Root access to real or virtual hosts in your private data centre or cloud service provider for your origin server and points of presence (PoPs)
2. Varnish instances with the following components:

a. **Massive Storage Engine (MSE)**

b. **Varnish High Availability (VHA)**

c. **TLS/SSL Encryption**

d. **Varnish Total Encryption**

e. **Varnish WAF**

a. Massive Storage Engine (MSE)

The Massive Storage Engine (MSE) provides persistent caching for hundreds of terabytes, and can serve several gigabytes of data per second. MSE uses an extent mapped file system in user space as the storage medium for the cache. This file system is in practice created as a file using the mkfs.mse utility. The following snippet shows how to create the file system for a persistent storage of 5GB.

```
$ mkfs.mse -s /var/lib/varnish/mse/
store,5GB \
-b /var/lib/varnish/mse/book,50MB
Creating data stores in file '/var/
varnish/mse/store'
Cooking the books in file '/var/lib/
varnish/mse/book'. This may take
a while ...
Creating 209661 spare nodes
Creating 209661 spare nodes
Finished
Varnish Plus Massive Storage Engine
data files successfully created.
Varnish Plus stevedore argument:
varnishd -s mse,
/var/lib/varnish/mse/store,/var/
lib/varnish/mse/book
```

b. Varnish High Availability (VHA)

The Varnish High Availability (VHA) component is a content replicator that adds resiliency and increases performance for high-demand cached content. The VHA agent triggers cache replication into its paired Varnish servers every time a new object enters the cache. As a result, objects can be cached and retrieved from multiple servers with only one backend request. With VHA, you also ensure consistency among different caching nodes and consume less bandwidth.

The creation of VHA nodes is as simple as listing the nodes to replicate in a plain-text configuration file

```
/etc/vha-agent/nodes.conf:
singapore-01 = https://cdn-
singapore-01.varnish-software.com
singapore-02 = https://cdn-
singapore-02.varnish-software.com
```

You then configure specific options as when you use self-signed certificates in /etc/sysconfig/vha-agent.params (or where your vha-agent file lives) and finally create the VCL code by running the vha-generate-vcl.

As a last step, you include the vha.vcl in your VCL file:

```
include "vha.vcl";
```

c. TLS/SSL Encryption

Varnish supports incoming and outgoing TLS/SSL encrypted traffic. Encrypted data entering Varnish is decrypted using the integrated component Hitch, which is designed to handle tens of thousands of concurrent connections efficiently on multicore machines, using hundreds of thousands of certificates. Outgoing traffic is encrypted by simply toggling configuration parameters in Varnish servers, as illustrated in this snippet:

```
backend default {
.host = "host.name";
.port = "https"; # This defaults to
https when SSL
.ssl = 1; # Turns on SSL support
.ssl_sni = 1; # Use SNI extension
.ssl_verify_peer = 1; # Verify the
peer's certificate chain
.ssl_verify_host = 1; # Verify the
host name in the peer's certificate
}
```

d. Varnish Total Encryption (VTE)

Varnish Total Encryption is an optional security feature that can encrypt the entire cache. It encrypts each and every cached object with its own unique AES256 encryption key. With VCL, you can decide which objects need to be encrypted and which don't. Again, it's flexible to meet your own security needs. The total encryption option helps keep unwanted eyes off cached content, making it unreadable, but also prevents cache leaks, such as Cloudbleed and Meltdown.

Varnish Total Encryption is a Varnish module (VMOD) in VCL that works with all Varnish cache storage types including malloc, MSE and MSE with persistence. Using Varnish Total Encryption requires the addition of just one line of VCL:

```
include "total-encryption/random_
key.vcl";
```

The flexibility of VCL makes it possible to use Varnish Total Encryption to create larger secured architectures, such as for your own CDN.

e. Varnish Web Application Firewall (WAF)

Varnish Web Application Firewall allows you to set your own security rules in ModSecurity style. It is implemented as a VMOD, making it configurable in VCL.

The Varnish Web Application Firewall (WAF) lets you inspect your HTTP traffic and detect malicious requests at the edge before they reach your web application. This lets you prevent code injections, malicious clients and protect your origin servers. It can be considered a security perimeter defence.

5. Configuration

Implementing Varnish servers makes the construction of your private network simple, yet powerfully configurable.

To have a basic and working configuration you need to:

1. Let Varnish know which backends it can talk to: Open "/etc/varnish/default.vcl" and specify your backend IP address and port
2. Configure which port on which Varnish listens to incoming request: Open "etc/default/varnish.param" and set "-a": 80

Restart Varnish, and you are all set. You have your very own CDN up and running. Once you have understood what type of infrastructure best meets your needs and how your audience is distributed, following these steps to build your own private CDN using Varnish should only take a few hours.

Conclusion –

Content delivery is undergoing a major transformation with everything moving to high-resolution and interactive media. The challenge from this perspective is that content providers cannot scale revenue linearly with the traffic growth. For example when moving from SD (standard definition) delivery to HD (high definition) delivery, the number of ad impressions (pre-rolls, commercial breaks, and post-rolls) is not increasing. Or, for premium publishers the subscription fee does not go up. For public broadcasters the license revenue does not increase.

However, their traffic demand significantly increases, which impacts content delivery cost, which will further accelerate with the introduction of new even higher quality formats such as 4K and 8K and new interactive experiences, such as virtual reality and 360 videos.

Content providers need flexible tools to meet the boom in user demand for high-performing, instant content delivery. With Varnish, content providers can build their own solution with Varnish Private CDN and get exactly this flexibility and performance as well as the ability to choose the form and method of content delivery that meets the unique needs of their content and audiences. Take back control of how, when, to where, from where and to whom you deliver content - and at what cost.

www.varnish-software.com

Los Angeles - Paris - London

Stockholm - Singapore - Karlstad

Dusseldorf - Oslo - Tokyo

www.varnish-software.com