

[netlify.com](https://www.netlify.com)

To WWW or not WWW | Netlify

Note, if you use Netlify's built-in [DNS service](#), there's absolutely no difference between using `www` or not - if you're using an external DNS provider, then read on.

Your DNS Setup can have a great effect on performance and DDoS resilience. Our site for Netlify is hosted on `www.netlify.com`, rather than `netlify.com`

We call `netlify.com` the “apex” domain or “root” domain.

Some might think that this is merely a cosmetic difference and make whole websites [arguing against the `www` prefix](#). However, dropping `www` from your main site domain can have dire consequences because of how DNS records work.

DNS Records

DNS is the internet-wide phonebook that tells browsers (and any other connected system), how to resolve a specific domain name to an IP address of a server they can open a connection to.

For websites, there are 2 kinds of records relevant to hosting a website with Netlify:

- **A Record** An A record returns an IPv4 address of a server
- **CNAME Record** Returns another domain name that the browser should lookup instead

When you configure a custom domain to point to a Netlify site, we always recommend that you use a CNAME record pointing at `<yoursite>.netlify.com` (where `<yoursite>` depends on the name of

your Netlify site).

Let's say you've configured `www.example.com` to return a CNAME pointing at `example.netlify.com`.

When a user visits your site, her browser will lookup the DNS record for `www.example.com`, it will receive a CNAME telling it to look up `example.netlify.com` instead. When it looks up `example.netlify.com`, it connects to our advanced traffic director, that returns an A record with an IP address of the server from our pool of currently available CDN nodes that's geographically closest to the end user.

The cool thing about this is that if our system has detected that our main load balancer is currently being hit by a large DDoS attack and is slow or unresponsive, we'll simply route around that on the DNS level. Because we cache content at our edge nodes around the world, end users also experience extremely fast page load times.

The Trouble with Apex Domains

This is all great, but let's say you'd like your site to live directly at `example.com` instead of `www.example.com`.

You might think you could simply configure a CNAME record for the apex domain and be done with it, but you shouldn't! This could severely break other things for you.

The problem with CNAME records is that the DNS specification doesn't allow any other DNS records on a name that has a CNAME record set.

When you have a specific name for your website like `www` or `app` or any other subdomain, that's not a problem. For the apex domain, however, this is different.

If you want to be able to receive email on your domain, you'll need to set MX records at the apex domain. With a CNAME, no other records can be set.

Want to validate your domain for something like [Google's webmaster tools](#)? Or perhaps you'd like to use the [SPF system](#) to help verify your email sending hosts? Now you have to add a TXT record to the apex domain. If you already have a CNAME, again, that's not allowed.

A records, Outages and DDoS Attacks

Even if your site lives at `www.example.com` instead of `example.com`, you'll still want a way to point `example.com` at our servers so we can serve a 301 redirect from `example.com` to `www.example.com` when a user visits the apex domain.

Because setting a CNAME record on the apex domain will break email and wreak havoc on TXT records, the only remaining option is currently an A record (in the future AAAA records will be an option as well).

Because an A record needs to point to a raw IP address, we publish the IP address of our main load balancer for this purpose.

The problem is that an A record doesn't let us insert any traffic direction between the end user's DNS lookup and our infrastructure. So we have no way of routing users to the closest CDN node, and if our main load balancer for some reason goes unresponsive, we have no way of routing traffic coming directly from A records around this.

Therefore, we **strongly recommend** that you always host your main site or app on `www` or any other subdomain (`app.example.com` is perfectly fine as well). This way you can use a CNAME record for the canonical site URL. You'll take full advantage of our globally distributed infrastructure and we can automatically route around any localized outage.

The root domain at `example.com` will still work, but will simply redirect to `www.example.com`. With Netlify this redirect happens straight on the CDN edge nodes and is extremely fast.

CNAME Flattening, ANAME or ALIAS Records

However, there is an available alternative! Some DNS hosts have paid attention to the `no-www` crowd and implemented a workaround.

NS1 offers [ALIAS records](#), DNS Made Easy has support for [ANAME records](#) and Cloudflare supports [CNAME Flattening](#).

None of these is part of the DNS spec, but are all different workarounds to solve the limitations of CNAMEs.

When you set an `ALIAS` record for `example.com` pointing at `example.netlify.com`, then NS1 will do the CNAME look up itself, instead of delegating that to the browser.

Normally the browser would get a CNAME record, follow that to `example.netlify.com`, and then receive the IP address of the closest available server.

With an ALIAS record, instead NS1 itself checks `example.netlify.com` and returns an A record with an IP address directly to the browser.

There are two benefits:

- **You essentially get the benefit of a CNAME for your apex domain**
We can still route around outages and do some geographic traffic direction
- **End users save an extra DNS lookup** Instead of first looking up the CNAME record, and then looking up the final IP, the browser gets an IP straight away.

However, there's also an important drawback:

- **The precision of the geographic lookups that Netlify does might suffer** We now do our lookup based on the IP of the DNS server rather than the IP of the end user, which can lead to inaccuracies based on the lack of

granularity of those servers. Plus, for performance reasons, DNS servers can do some caching of IP lookups that can lower the precision. In general these drawbacks are not severe or universal in their effects, so given the choice, the enhanced record should be used instead of the A record. But, feel free to [ask support to take a look at your setup to advise](#).

The ideal setup

The ideal setup that we recommend is to use `www` or any other subdomain as the canonical domain for your website, but still use a DNS host that supports some form of CNAME on the apex domain. [Our own DNS hosting](#) does this, for instance (and handles configuration automatically for you too!)

This way you'll get the full benefit of our global DNS traffic direction on your main site, and we can also route around outages when doing the redirect from `apex` domain to `www` domain.

Hope this was informative!

About Netlify

Netlify is an all-in-one platform for deploying and automating modern web projects.

Simply push and Netlify provides everything—servers, CDN, continuous delivery, HTTPS, staging environments, prerendering, asset post processing, DNS, and more.

Any project, big or small, can perform instantly on a global scale.