Web Domain Name

After purchasing a domain name, we can configure SWAG to generate a secure certificate that ensures our server's security and authenticity.

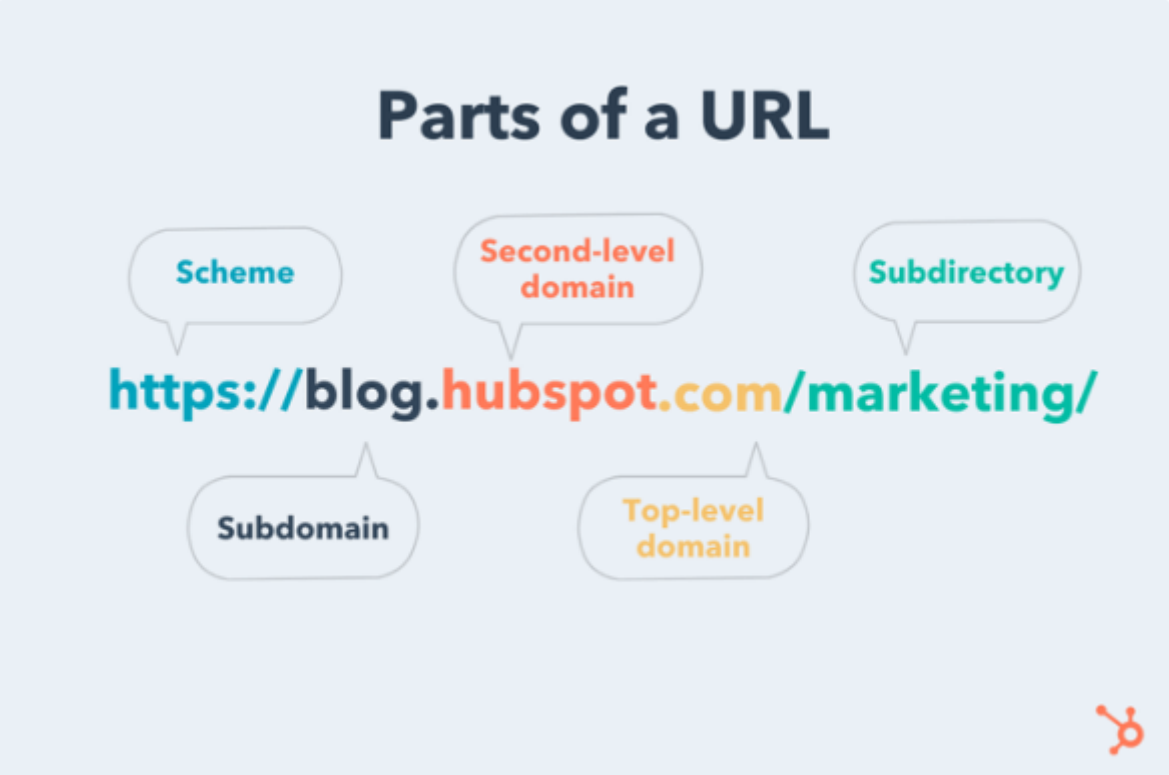
Most people will only need a single domain address with sub-domains used for each of our web services. If you are trying to run multiple different brands, you can also host multiple domains from the same web server.

- Domains & URLs
- Getting a Domain Name
- Domain Name System
- Reverse Proxy
- Connecting Services to the Reverse Proxy

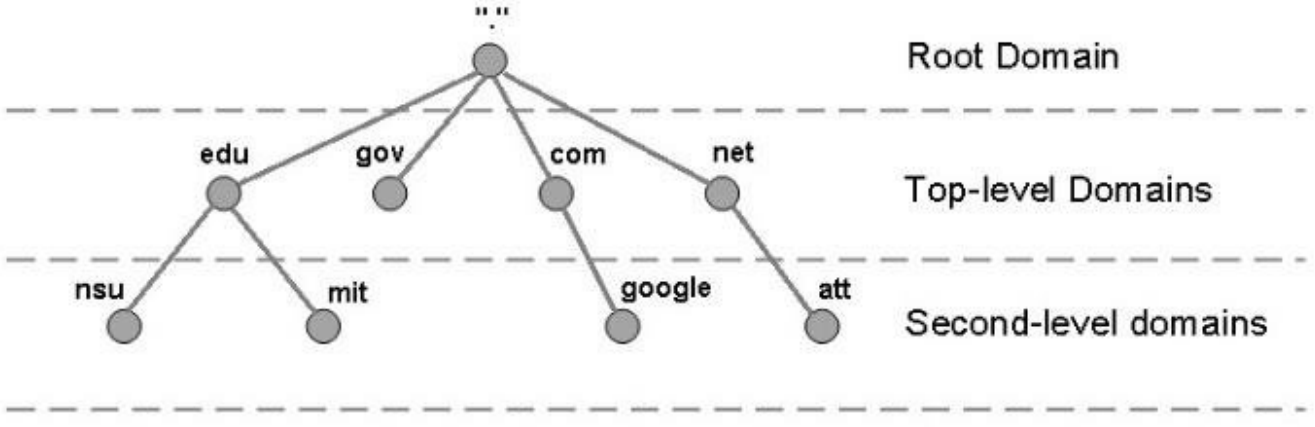
Domains & URLs

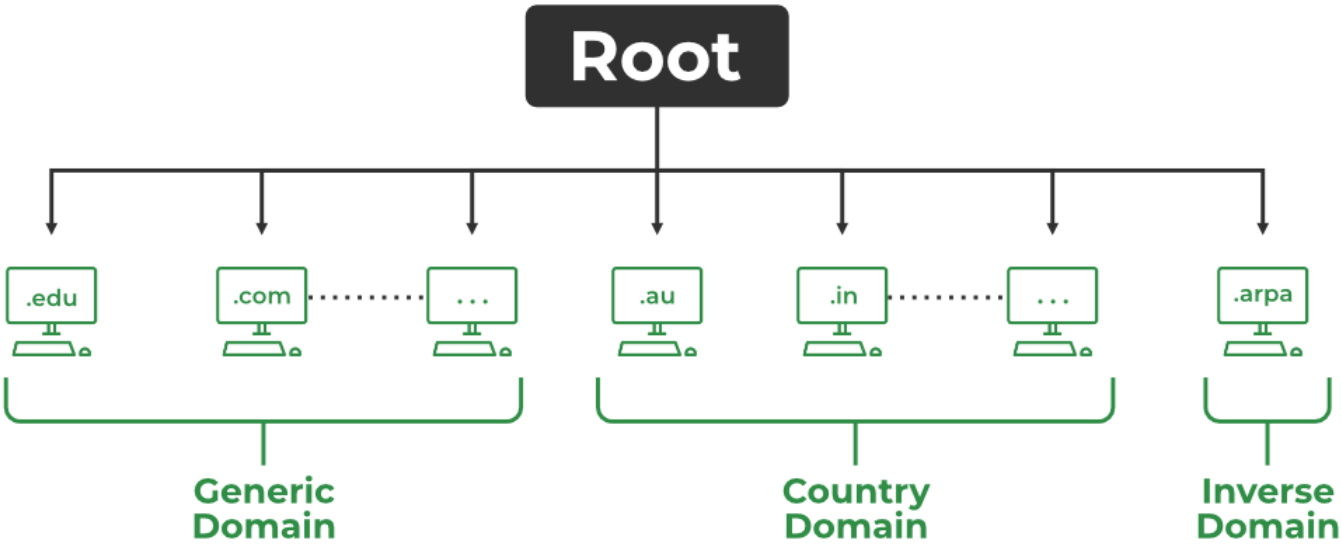
When navigating the World Wide Web through a browser, we enter an address in the form of a URL - or Universal Resource Locator – that contain the information necessary for your computer to connect to another located anywhere in the globe. Each URL has a few core components that help specify a location and how to reach it.

> A full web domain URL might be https://www.example.com.



Domain names are used to identify autonomous areas of the World Wide Web under the control of a person, corporation or other entity. These are used for gaining access to websites, email and other services. Web domains are listed on the Domain Name System – a distributed registry to help locate servers around the globe.





Much like the file system on the Linux operating system, Web addresses start from a root that contains all domains. Stemming from the root, there are top-level domains like the popular .com, .info, .net, .edu, and .org. There are also special country code web domains, as well as the recently expanded purpose-specific domains – such as .news, .tech, .name or .blue.

These "TLDs" are carefully maintained by the Internet Corporation for Assigned Names and Numbers (ICANN) – a global non-profit that handles the development of policies and procedures for the Domain Name System.

Second-level domains offer personality and customization to a URL. They are combined with a top-level domain using a period to create a full web domain name. When purchasing a domain name, you are generally renting access to this specific combination of top- and second-level domain for a pre-determined period of time.

> Domain registrars often provide discounts for the first year, but following years come with higher annual renewal fees.

Single Domain

For many people, you will probably only require one domain name. We will be using SWAG to connect our server to the World Wide Web and this project is configured to use a single domain with sub-domains as a default.

Sub-Domains

When offering multiple independent services from your server – such as OwnCloud or Radarr – we can use sub-domains to separate these web applications into different URLs.

> Using this example, radarr.example.com could direct you to the Radarr web application.

These are an excellent way to build a digital community and brand identity around the same domain name. With this technique, we could host a primary website at example.com, as well as a Flarum forum at forum.example.com and a WordPress blog available at blog.example.com.

Multiple Domains

SWAG can configured to act as the access point for multiple different web domain names – such as example.com and example.org. This makes it so you can run a personal and professional web domain from home using the same server.

> Each of these domains will be listed under the same SSL certificate and therefore linked.

Getting a Domain Name

Before you can connect your server to the World Wide Web and access it through a Web browser, you will need a domain name. This address will be used to generate the SSL certificate that web browsers rely on to create a secure network connection and verify we are who we say we are.

Domain Registrars

While ICANN – an American 501(c)3 non-profit located in California – is tasked with the development of the global Internet infrastructure and security, they do not have the capacity to orchestrate over 350 million global web domain names. Instead, over 2,800 domain name registrars have been accredited to operate in their capacity.

This process requires ongoing adherence to strict regulations including additional clearance required to offer special or country-specific domains. These rules, set by ICANN, require that the ownership of domains can be transferred between registrars. They set explicit pricing restrictions on some top-level domains – such as .com and .net – while also imposing a maximum domain registration period of ten years.

While the ICANN has requirements for domain registrars, they are not all created equal. Some employ annoying marketing strategies while others have lax security or even actively malicious practices. Aside from select pricing restrictions that are required for accreditation, domain registrars are free to operate their service however they desire. This includes pricing schemes that reflects expected market desirability and popularity – such as home.tech costing $650,000.

Domain registrars are required to provide information to WHOIS – a public-access database about domains, including contact information for the person who owns the right to it. Registrars commonly offer privacy services that withhold personal identifying information.

> You don't need to pay for an SSL certificate through the domain registrar because we will be generating them for free using SWAG.

Cloudflare

Based in America, this company provides cybersecurity services to nearly 20% of websites, including a free consumer tier. They are also an accredited registrar that provides at-cost domain name services for most top-level domains.

Micro.Domains

This service leverages Namecheap's infrastructure to sell domain names that are 5-characters or less. While these domain names are often random, they offer marginal 'security by obscurity' for accessing personal web services at a reasonable and transparent price.

PorkBun

This service is operated by the American business Top Level Design, offering an intuitive experience and transparent rates for domains down to $2.

NameSilo

This service is operated out of America and focuses on customer privacy. They offer transparent pricing and accept many different payment options.

Dynamic DNS

Most consumer Internet Service Providers assign Public IP addresses to residential networks that can change at any time. Proprietary Dynamic DNS services offer tools for consumers to quickly update their Public IP address and make sure their server is always accessible. There are several free services available, but they come with drawbacks.

> SWAG can generate a working SSL certificate to ensure data privacy, but browsers may not be able to verify this certificate and can lead to browser-based warnings.

DuckDNS

This free service offers a free sub-domain under duckdns.org – such as example.duckdns.org. They use Amazon servers, but can be easily integrated into your Web server setup.

DDclient

This open-source utility makes it simple to keep one or more Dynamic DNS services up-to-date with your current Public IP address. This will require more initial setup, but it can create redundancy through multiple services. DDclient currently supports over 30 different dynamic DNS services. During the creation of an SSL certificate, you will need to manually validate your server using the HTTP mechanism provided by SWAG.

Manufacturer DDNS

Consumer routers from mainstream technology manufacturers – such as ASUS, TP-Link and Netgear – are providing built-in Dynamic DNS features. This will provide a sub-domain under their dynamic domain service. During the creation of an SSL certificate, you will need to manually validate your server using the HTTP mechanism provided by SWAG.

Domain Name System

We need to configure a DNS service to handle the translation of our domain name into the Public IP address provided by your Internet Service Provider. There are three ways to accomplish this: one requiring a paid domain name, while the two others are free subscription services.

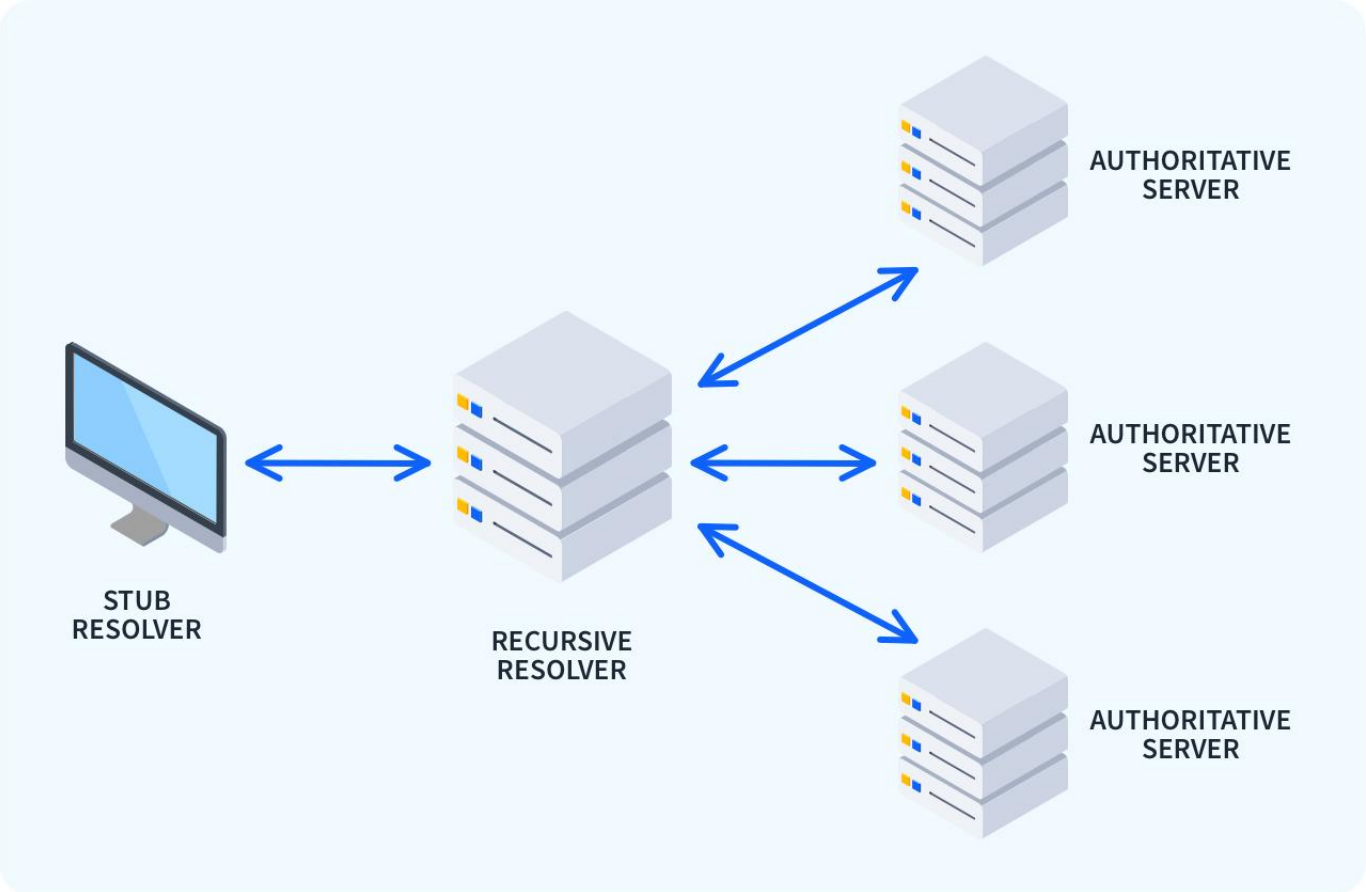
Using Cloudflare

> This is the recommended method.

Cloudflare is a content delivery and cybersecurity services company that offers free basic-tier solutions for anyone running a web service. We will be using their DNS service and nameservers to direct traffic to our server. They offer protection from Distributed Denial of Service – more commonly known as DDoS – attacks as well as defensive tools in the event that you are targeted. They will provide another layer of security to our web services.

Nameservers

When purchasing a domain name through domain registrar, they will generally to use that registrar's web services. This includes an authoritative nameserver that serves as a directory of domain names they are providing service for. The Domain Name System powering the internet is decentralized and no one entity owns it explicitly. A recursive resolver is used to systematically search these disparate nameservers and find the desired domain.



We will need to configure our domain name provider to use the CloudFlare nameservers. This will enable us to leverage their free services. The process to configure your domain's nameserver will be different based on the registrar you used. We use redundant servers to ensure that at least one is always available even if their are outages.

> When purchasing a domain through Cloudflare, they are pre-configured to utilize their nameserver and security services.

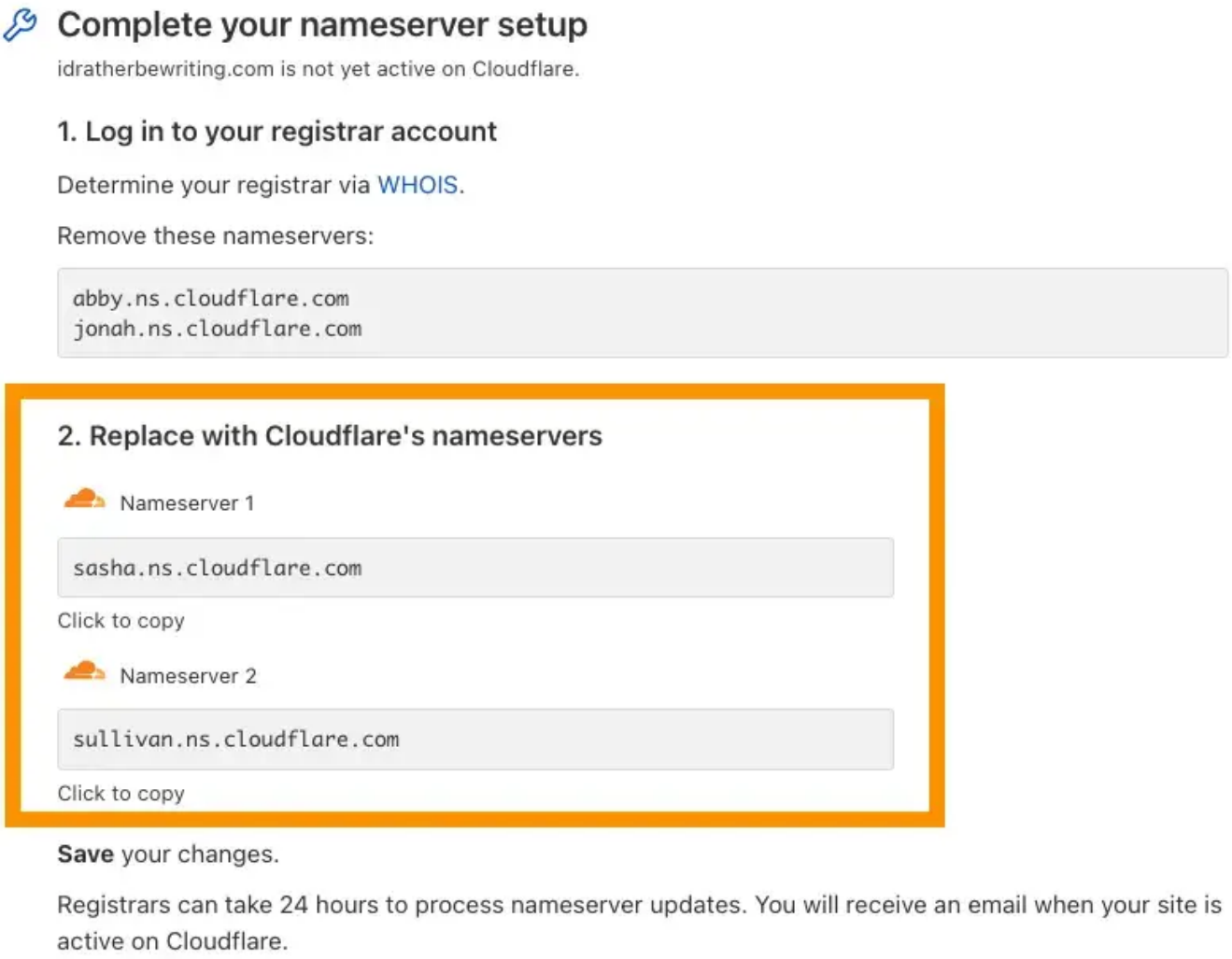
These can provide insights for a select few domain name registrars:

- NameSilo
- Porkbun
- NameCheap

We will need to use the Cloudflare nameservers to leverage their services. They host an assortment of decentralized nameservers to split up the workload.

> You will need to create a Cloudflare account. If you want privacy and anonymity, ProtonMail allows you to create separate email aliases.

When creating an account through Cloudflare, we will first need to add our site to their dashboard and then they will assign you two nameservers.



> Nameservers generally update quickly (every ~15 minutes) but it may take up to 24 hours.

You will receive an email notification once this process is complete. After you have added their nameservers through your domain name registrar, you can complete the Cloudflare domain verification.

> For enhanced security, you should follow the Cloudflare guide for enabling DNSSEC.

DNS Records

Once our domain is configured to use Cloudflare's nameserver, we will need to configure traffic received at our domain to be directed to the server located at our Public IP address. DNS records – much like a label on a filing cabinet folder – explains what can be found within. These can also be used to configure email addresses, social media handles or even store public notes.

If you pay your Internet Service Provider for a static Public IP address, you can direct the domain to your server and you'd be done. We need to create an A Record to direct traffic to our IP address using the Cloudflare dashboard.

This record should have the name '@' to signify we are setting the IP address for the root of our server – such as example.com – as opposed to a sub-domain. For the IP address, we need to add the Public IP address provided by your Internet Service Provider. If you are unsure, you can view your public IPv4 address by visiting a web service like What Is My IP?.

Dynamic Addresses

Most residential Internet service plans do not come with a Static Public IP address by default. This is generally restricted to commercial business internet plans for an additional fee. Home Internet connections generally use dynamic IP addresses that may change at any time.

There are open-source software options to automatically update our IP address within DNS records. LinuxServer.io maintains a Docker image for ddclient which can connect to CloudFlare through their API to ensure the IP address is always accurate. This will require creating an API key for Cloudflare, installing ddclient and configuring it with a text editor.

Install ddclient keyboard_arrow_right

Using DuckDNS

DuckDNS is a free and proprietary service where you can reserve a sub-domain – such as example.duckdns.org. This domain can be directed to your server and configured to work with your individual services. You will need to make an account with the service by logging in using to Google or GitHub through their homepage.

Once you have logged in, you can register your sub-domain through the service and assign your IP address to it. On your account page, there is a private token that can be used to automatically update your IP using a DuckDNS Docker image.

Install DuckDNS keyboard_arrow_right

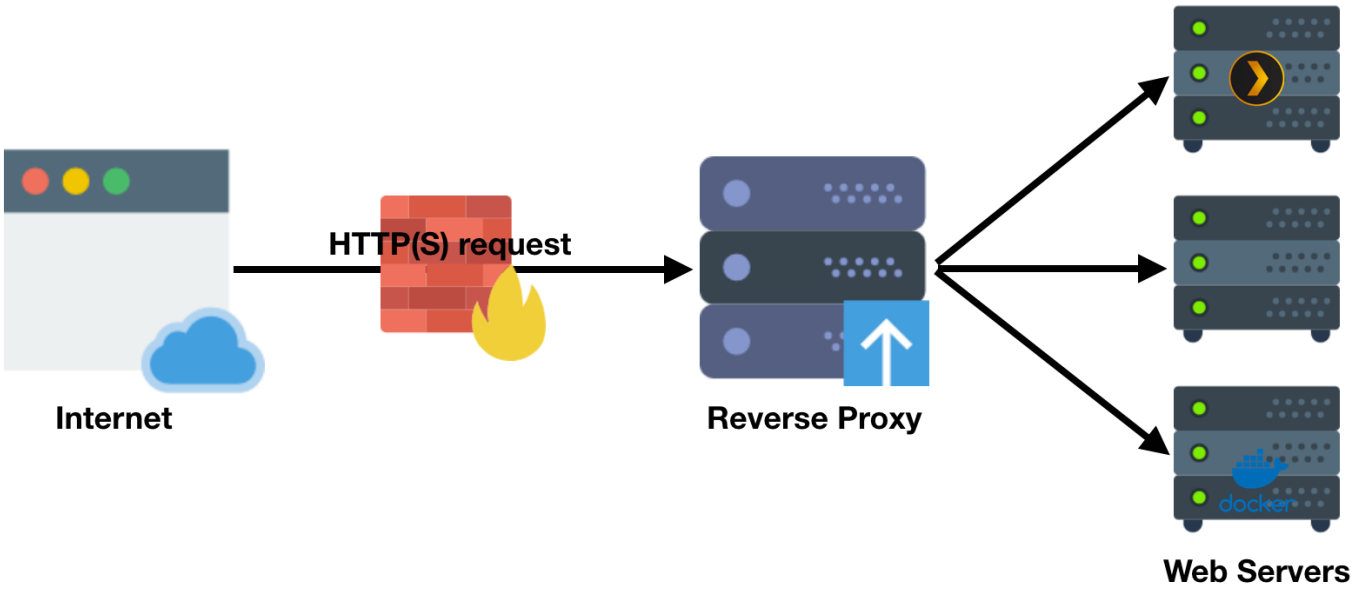
Manufacturer Dynamic DNS

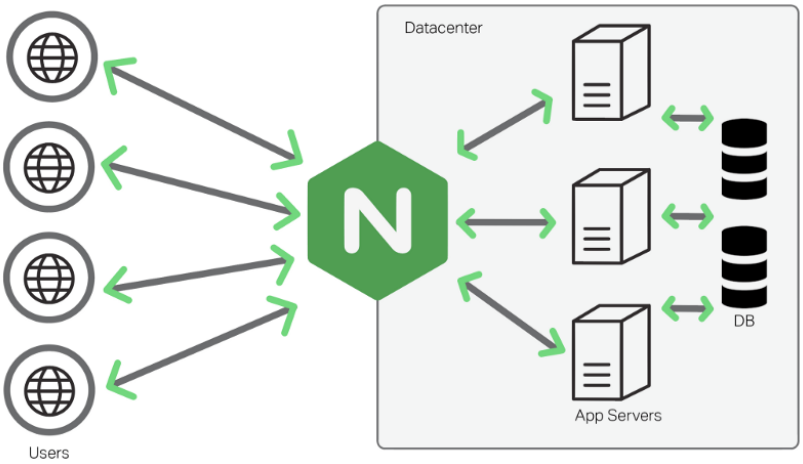
Consumer routers from mainstream technology manufacturers – such as ASUS, TP-Link and Netgear – are providing built-in Dynamic DNS features. By creating an account with them, you can access a sub-domain under their dynamic domain service. this will happen automatically without intervention.

Reverse Proxy

We need to install a reverse proxy to safely access our web server over the World Wide Web. This specialized server software sits in front of other servers and retrieves websites on behalf of the people trying to access them. They act as the public-facing front for a hidden network of computer servers operating behind-the-scenes on your local area network.

When passing along web site data, a reverse proxy will actively overwrite any information about the server it came from. During this process, they can also modify the HTTP headers used to silently communicate information between a web server and browser. This also can be used to inject information into the stream, changing the way a website looks and operates from the top-down.



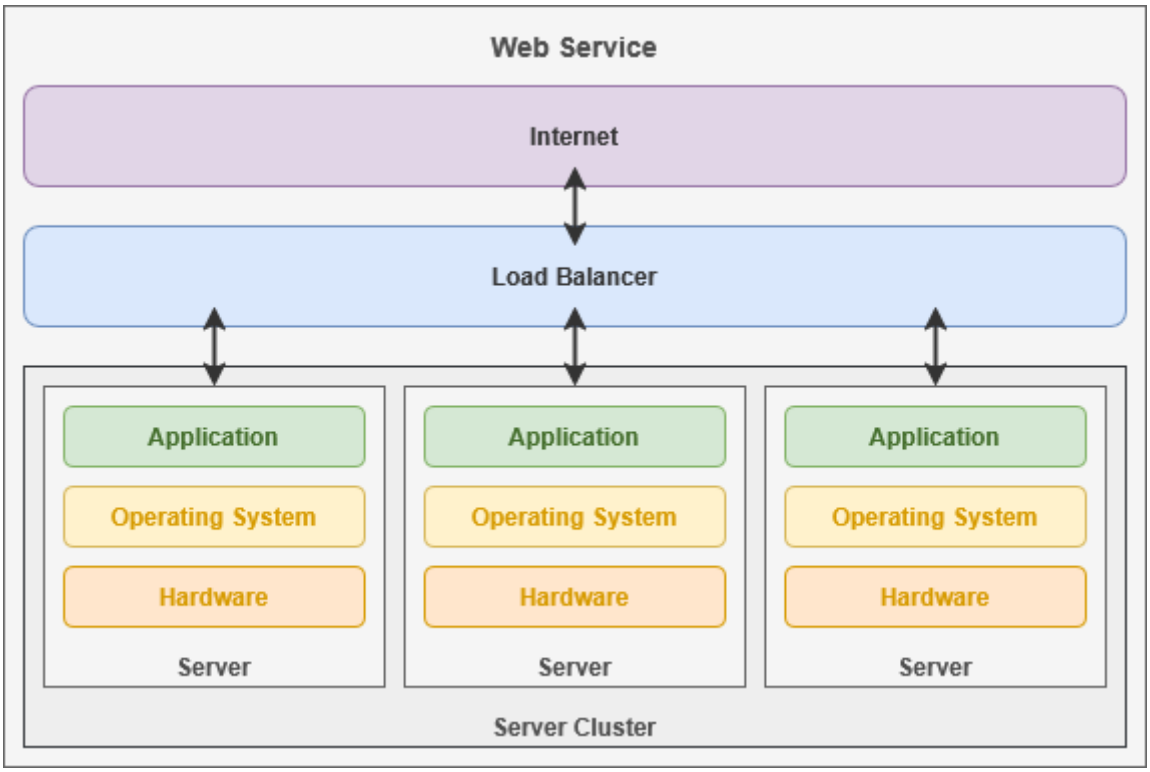


This lets us keep access to our services behind the privacy of our Local Area Network and ensures no one can ever communicate directly with an origin server. This decreases our attack surface – and improves overall security – by only giving the Internet access to the reverse proxy instead of

by individual service. In the event that our server is hacked, they only gain access to the reverse proxy and not any of our underlying services. However, a compromised reverse proxy has the potential to cause a great deal of damage and should be safe-guarded.

{{Show difference between reverse proxy and connecting individual services to the internet.}}

For cloud computing environments, reverse proxies provide powerful speed benefits on multiple fronts. Acting as a cache – or a copy of frequently requested data – a reverse proxy can step in and take the strain off of individual services by handling simple requests. They can also work as a load balancer to spread out users across multiple independent servers. This is how large cloud websites provide access to millions of users.



Encrypting data for secure transmission over the open internet can require a great deal of hardware resources. A reverse proxy can be indispensable because they can handle the encryption (and decryption) of secure data with clients outside of the Local Area Network. This stream can also be compressed so that less data needs to be transmitted over the internet.

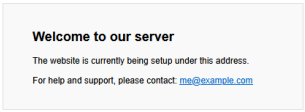
By acting as the singular access point for a plethora of behind-the-scenes services, they can also represent a single point of failure where one malfunction can leave you without any access. This is a contingency that we prepared for by setting up local network access protocols, but it might be prudent to operate your web server in a place where you can access a mouse and keyboard.

Secure Web Application Gateway

Secure Web Application Gateway – more commonly referred to as SWAG – is a community-driven project by LinuxServer.io to host a secure and easy-to-use reverse proxy and web server.

Nginx ("engine x") is an open-source HTTP server, reverse proxy, web cache and load balancer that is used to power the majority of corporate domains. This is the core of the SWAG self-hosting project – including pre-configured add-ons and templates for accessing popular self-hosted services behind a reverse proxy. We will be configuring this to access our individual services from a centralized location.

Nginx can be used to host an HTTP website with full PHP functionality and add-on modules for accomplishing specific tasks. By default, SWAG will host a basic website until we configure our reverse proxy to access our self-hosted services.



When we use the Internet to connect to our SWAG container running in Docker, we are connecting through an HTTP (port 80) or secure HTTPS connection (port 443). When the reverse proxy receives traffic, it will inspect the data to see where and how it should be delivered. When accessing the website through the primary domain – such as example.com – the traffic will be transmitted to the self-hosted website.

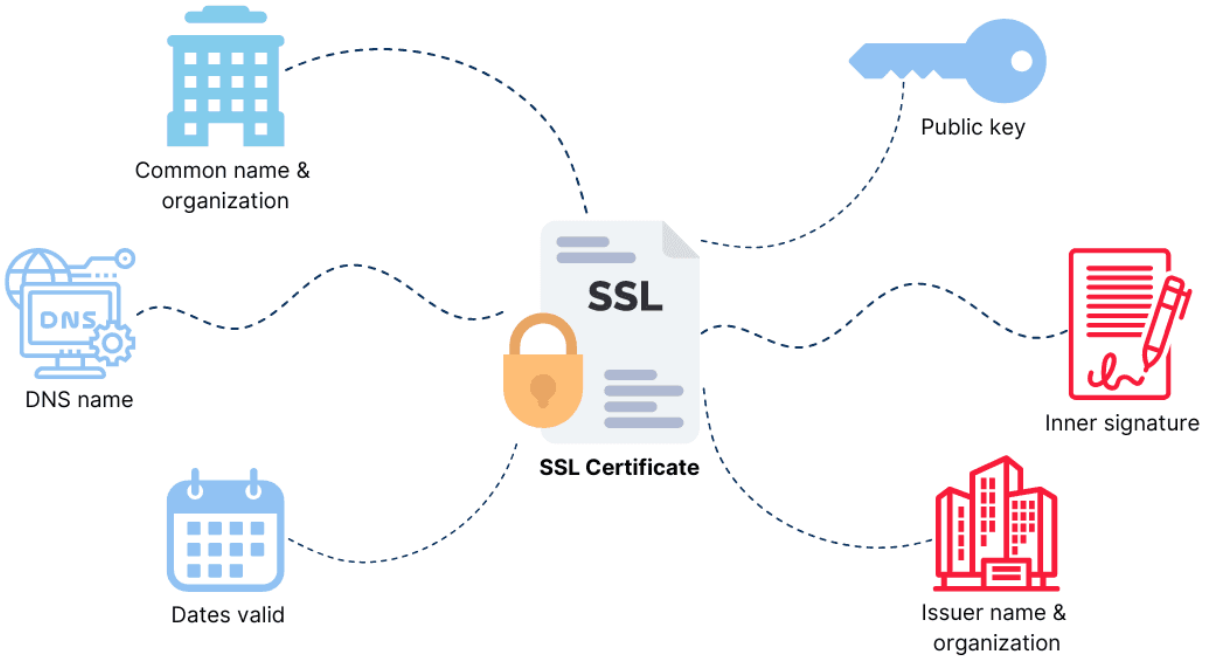
When accessing a sub-domain – such as jellyfin.example.com – the data will be transferred to the Jellyfin server in order to handle the client's request. All data transmitted between the reverse proxy server and Jellyfin is hidden from the World Wide Web and protected behind our Local Area Network.

> A reverse proxy forwards a website as it is – meaning that you need to ensure security through encryption and password-protection before connecting it to the Internet.

We are hosting services using our server and attacing their web interface to local ports that are accessible to other computers on our local network. For example, this is how we access Cockpit at its default port 9090 through a Web browser. By using a reverse proxy, we can route this same access through a web sub-domain – such as cockpit.example.com.

> It is recommended that services like Cockpit are restricted to Local Area Network access.

SWAG makes it easy to automatically generate an SSL certificate using a variety of mechanisms. These form the foundation of the HTTPS protocol by verifying the identity of the server and encrypting the data sent through a secure TCP connection.



SWAG also provides pre-configured settings for integration with other security-focused add-ons:

Language

Geo-Fencing Services

This service offers the ability to either block – or selectively allow – clients seeking access from specific geographical locations.

SWAG comes with accessible defaults that balance security and convenience. The software it's built on is fully open-source and that means you can configure it however you need – even if that means modifying the underlying software.

The repository includes community-tested templates for each of the services available here, as well as many more. This will require first configuring and installing the software through Docker Compose using Portainer.

Installing SWAG keyboard_arrow_right

Connecting Services to the Reverse Proxy

How to edit the SWAG configuration files to connect a service to the internet using the pre-made templates.