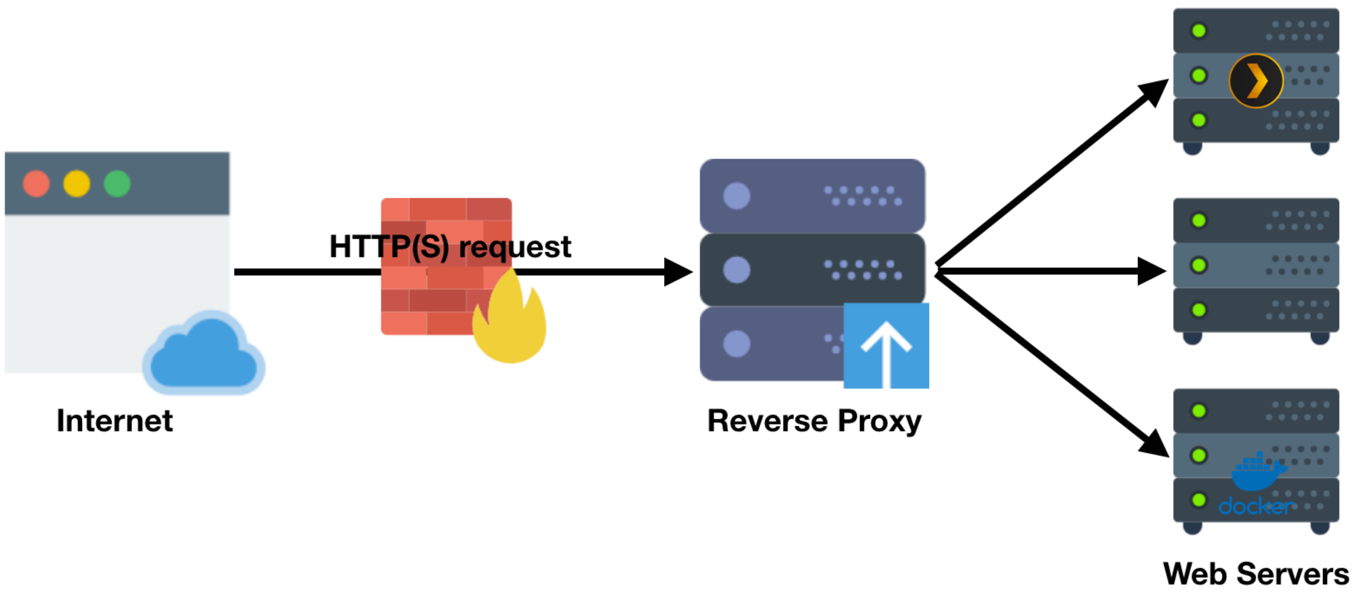
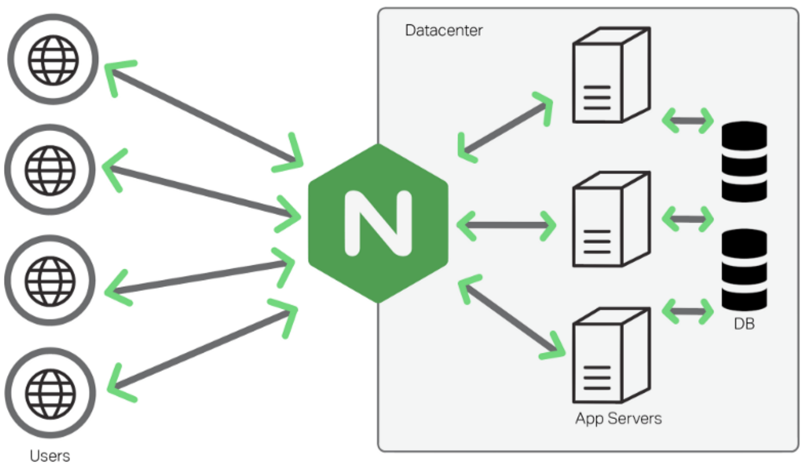
Reverse Proxy

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.



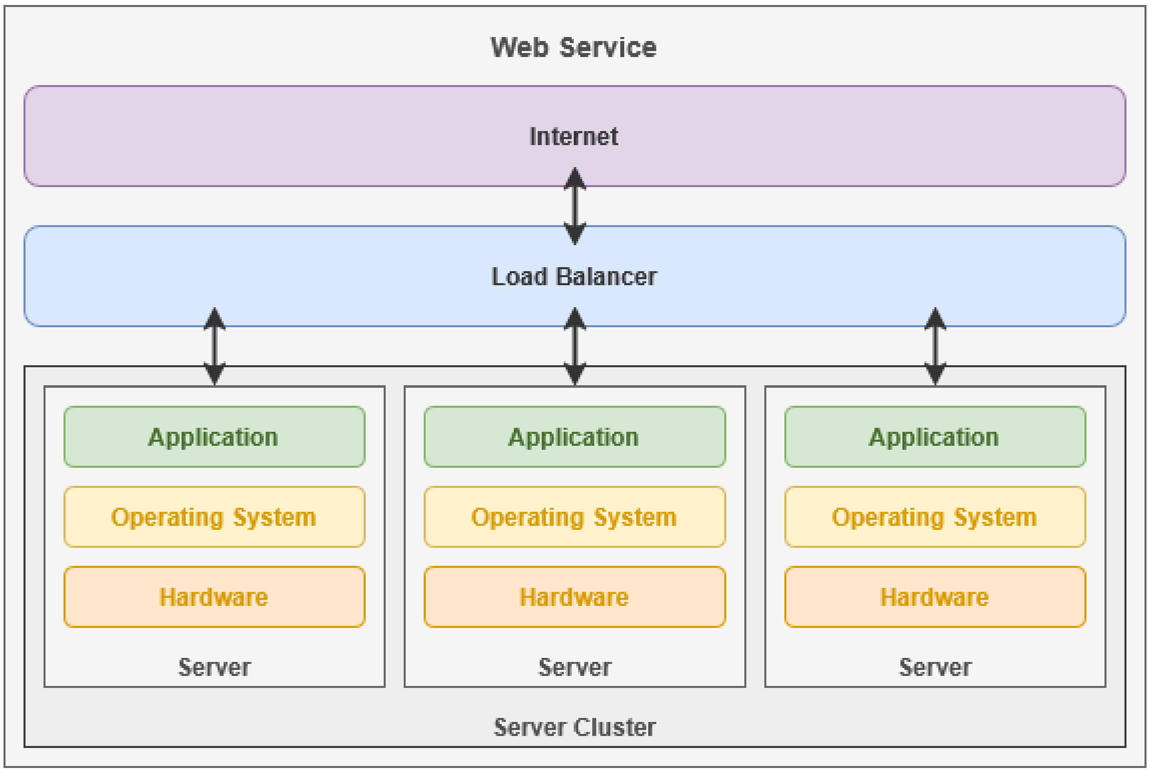


This lets us keep access to our services behind the privacy of our Local Area Network and ensures no one can ever communicate directly with an origin server. This decreases our attack surface –

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.

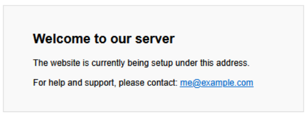
By acting as the singular access point for a plethora of behind-the-scenes services, they can also represent a single point of failure where one malfunction can leave you without any access. This is a contingency that we prepared for by setting up local network access protocols, but it might be prudent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


Welcome to our server

The website is currently being setup under this address.

For help and support, please contact: me@example.com

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.

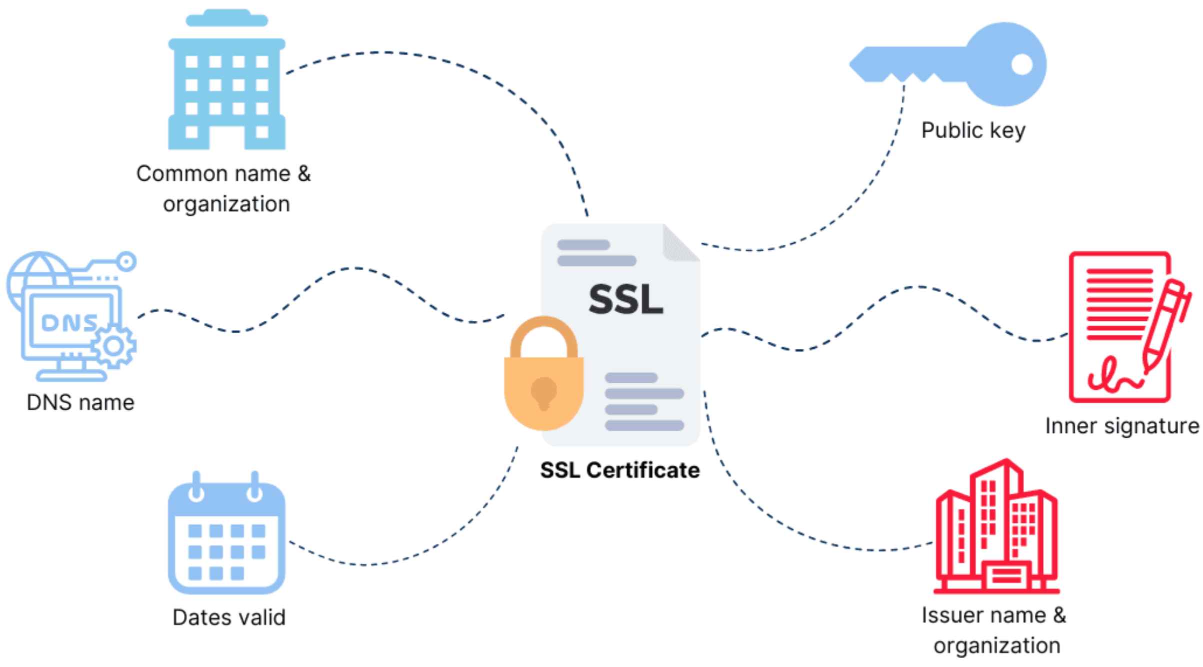
When accessing a sub-domain – such as jellyfin.example.com – the data will be transferred to the Jellyfin server in order to handle the client's request. All data transmitted between the reverse proxy server and Jellyfin is hidden from the World Wide Web and protected behind our Local Area Network.

> 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:

Dashboard

Dashboard
This provides a graphical overview of the common device types, geographical regions and IP addresses accessing your SWAG reverse proxy server.

Front_hand

Fail2Ban
This software offers intrusion detection that blocks malicious IP addresses that repeatedly fail authentication checks for your services.

Verified_user

Authelia
An open-source authorization portal that offers single sign-on and two-factor authentication for securing accessing your services.

Communities

CrowdSec
This projects offers proactive threat protection by fostering an open community to share information about malicious Internet actors.

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**

Revision #19
Created 29 April 2025 20:12:01 by metaphorraccoon
Updated 13 May 2025 07:29:37 by metaphorraccoon