Nginix

Overview

Nginix is a C based webserver commonly used as a reverse proxy to application servers. A reverse proxy is exposed to the internet on the standard HTTP(s) ports and is used to forward requests onto application servers, providing services such as:

* Load balancing – by sitting in front of your backend servers and distributing client requests across a group of servers in a manner that maximizes speed and capacity utilization while ensuring no one server is overloaded.
* Web acceleration –SSL, Caching, Compressing data
* Security and anonymity – acts as an additional defense against security attacks

How to Install

See repo

server\_name => domain names (e.g. cpcwood.com)

# Directives

<https://nginx.org/en/docs/http/ngx_http_core_module.html>

server

Defines a virtual server which normally has a listen directive to listne for requests on a specific ip and port. If ip is left out the default is used:

server {

listen: 80;

server\_name: example.com \*.example.com default\_server;

}

If there are several servers listening on the same port, nginx will test the Host header field against the server\_name directive. The server\_name can be an exact name, wildcard, or regular expression (Perl regex syntax, e.g. ~<regex>), matching the longest wildcard or first regex match in order of configuration file. The default\_server parameter can be added to the server\_name directive, to automatically define that server as default if no matches are made.

location

The location directive is used to set configuration based on the request uri. It is commonly used to serve specific files or send traffic to a different proxy. Nginx will test the request URI against the parameters of all location directives in a selected server, the one which matches, is that with the longest matching prefix.

root

The root directive specifies the root directory which will be used to search for a file. Nginx will append the request URI to the path specificed after the root directive. The root directory can be used within, http, server, and location contexts, therefore, different directories can be used when requests are matched to different areas of the application. For example, routing .mp3 files to a different storage directory:

server {

root /www/data;

location / {

}

location ~ \.mp3 {

root /www/media;

}

}

try\_files

try\_files is used to check whether a specific file or directory exists. It also allows for a default file or directory to be used, and custom error codes to be defined. Requests can also be redirected named locations if not found (e.g. proxy\_pass to an application server).

server {

root /www/data;

location /images/ {

try\_files $uri /images/default.gif;

}

location /homepage {

try\_files $uri $uri/ $uri.html =404;

}

location / {

try\_files $uri @backend;

}

location @backend {

proxy\_pass localhost:3000;

}

}

# Optimise

<https://www.arubacloud.com/tutorial/filter-and-optimize-static-file-requests-with-nginx-on-ubuntu-18-04.aspx>