

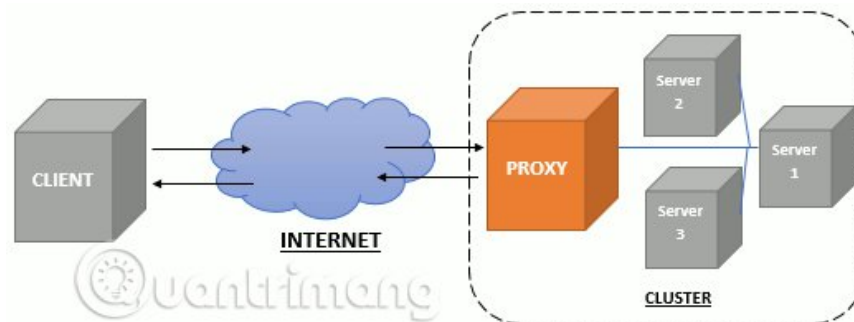
How to use Nginx as a reverse proxy

Unlike Apache, Nginx is the most popular web server available. In addition to being a web server, it can also be used as a load proxy or reverse proxy.

Unlike Apache, Nginx is the most popular web server available. In addition to being a web server, it can also be used as a load proxy or reverse proxy. In this tutorial, **TipsMake.com** will show you how to use Nginx as a reverse proxy.

What is reverse proxy?

A proxy server acts as an intermediary between the client and another server. Proxy server takes resources from the server you want to connect to and sends it to you to view. A reverse proxy works in the same way, except for a reversed role. When you request information from the server, the reverse proxy will keep the request and send it to the appropriate backend server. This allows the system administrator to use the server for multiple applications, as well as to ensure a smoother traffic flow between the client and the server.



Benefits of reverse proxy

Depending on your application, you may have different reverse proxy use cases.

1. **A reverse proxy allows you to run multiple applications on the same server** - If you have multiple applications running on the same server, all of them cannot 'listen' at the same time on ports 80 or 433. With reverse proxy, You can configure it to redirect traffic to individual applications as needed.
2. **Load balancing** - If there are multiple servers running the same application, you can use a reverse proxy to distribute traffic evenly to each server.
3. **Web application firewall** - You can use reverse proxy to hide your application and also to filter IP spam or protect from DDOS attack.

4. **Easy logging and monitoring** - Because all incoming traffic is managed by reverse proxies, logging and monitoring traffic flow will be easier.

Configure Nginx as a reverse proxy

To set up Nginx as a reverse proxy, the article will use the **proxy_pass** parameter in the Nginx configuration file.

Note : This tutorial assumes that you have some knowledge about Nginx and have it installed, as well as setting up Nginx in your server.

In most use cases, Nginx will be the front-end server, 'listen' to port 80 (HTTP) or 443 (HTTPS) for incoming requests. Since there can only be one 'listen' service on port 80 or 443, your application will have to 'listen' on another port, such as port 8081. The simplest configuration will look like this:

```
server { listen 80; listen [::]:80; server_name myapp.com; location / { proxy_pass
```

This means that all requests to **myapp.com** at port **80** will be redirected to port **8081**.

Advanced settings

In addition to directive **proxy_pass**, there are some other directives that you can use for more advanced settings.

1. **proxy_set_header** - This allows you to set the title to send to the background application. For example, see the following configuration:

```
proxy_set_header Host $host; proxy_set_header X-Real-IP $remote_addr; proxy_set_
```

You can set the proxy header needed to go to the application, tell it the required IP and remote address, then export the correct content to the requested website.

1. **proxy timeout** - This allows you to set the timeout value to send and receive proxy requests. For example:

```
proxy_connect_timeout 90; proxy_send_timeout 90; proxy_read_timeout 90;
```

1. **proxy_buffers** - This allows Nginx to temporarily keep the proxy server's response and only send it to the requesting server after the proxy server has finished responding. For example:

```
proxy_buffers 32 4k;
```

If your application sends a large number of files, you may want to disable **proxy_buffers**:

```
proxy_buffering off;
```

As you can see, Nginx is a reverse proxy server that has many uses. The biggest plus about it is that the configuration is simple, easy to use and still allows you to scale up in more complex situations. For more details, you can view the Nginx proxy module documentation (at https://nginx.org/en/docs/http/nginx_http_proxy_module.html) or configuration examples (at <https://www.nginx.com/resources/wiki/start/topics/examples/full/>).

Hope you are succesful.

You finished reading the article "**How to use Nginx as a reverse proxy**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.