

ABSTRACT

Web servers are a fundamental component of the World Wide Web, enabling the delivery of web content to users across the Internet.

A proxy server acts as an intermediary between clients and servers, facilitating communication and providing various functionalities in computer networks. This abstract provides an overview of proxy servers, including their definition, types, functionalities, and their significance in network infrastructure.

The proxy server is a server that acts on behalf of clients to communicate with other servers. It explains that when a client sends a request to access a resource, it is intercepted by the proxy server, which then forwards the request to the destination server. This intermediary role allows the proxy server to enhance network performance, security, and privacy.

Next, the abstract explores different types of proxy servers commonly used in networks. It discusses forward proxies, which handle client requests and relay them to destination servers. It also covers reverse proxies, which sit between servers and clients, distributing incoming requests and providing load balancing, caching, and SSL termination.

Furthermore, the abstract delves into the functionalities offered by proxy servers. It explains how proxy servers can cache frequently accessed resources, reducing the response time for subsequent requests. It also discusses how they can filter and block unwanted content, improving security and preventing access to malicious websites. Additionally, it touches upon the ability of proxy servers to anonymize client requests, protecting user privacy.

INTRODUCTION

1.What is web server

A web server is a software program that runs on a computer and allows it to serve up web pages and other content to users on the internet. It listens for requests from web browsers, such as Chrome or Firefox, and then sends back the requested files, such as HTML pages, images, or videos.

The web server software typically runs on a dedicated computer or server, and it may serve up one or many websites, depending on its configuration. Popular web server software includes Apache, Nginx, Microsoft IIS, and others.

Web servers play a critical role in delivering web content to users and enabling the internet as we know it. Without web servers, it would be impossible to access websites, search engines, online shopping, and many other online services.

A web server is a software program that serves content over the internet using HTTP (Hypertext Transfer Protocol). Its primary function is to listen for incoming requests from web browsers, retrieve the requested resources such as web pages, images, videos, and other files, and send them back to the requesting client

When you type a website's address (URL) in your web browser, the browser sends a request to the web server associated with that URL. The server then processes the request, retrieves the necessary files, and sends them back to the browser to display the website

Web servers can also provide additional functionalities such as SSL/TLS encryption, caching, load balancing, and server-side scripting using programming languages like PHP, Python, and JavaScript.

Overall, web servers play a crucial role in enabling the internet as we know it, by providing a platform for delivering web content to users around the world

1.1 How do web servers work?

Web server software is accessed through the domain names of websites and ensures the delivery of the site's content to the requesting user. The software side is also comprised of several components, with at least an HTTP server. The HTTP server is able to understand HTTP and URLs. As hardware, a web server is a computer that stores web server software and other files related to a website, such as HTML documents, images and JavaScript files.

When a web browser, like Google Chrome or Firefox, needs a file that's hosted on a web server, the browser will request the file by HTTP. When the request is received by the web server, the HTTP server will accept the request, find the content and send it back to the browser through HTTP. More specifically, when a browser requests a page from a web server, the process will follow a series of steps. First, a person will specify a URL in a web browser's address bar. The web

browser will then obtain the IP address of the domain name -- either translating the URL through DNS (Domain Name System) or by searching in its cache. This will bring the browser to a web server. The browser will then request the specific file from the web server by an HTTP request. The web server will respond, sending the browser the requested page, again, through HTTP. If the requested page does not exist or if something goes wrong, the web server will respond with an error message. The browser will then be able to display the webpage.

1.2 Examples of web server uses

Web servers often come as part of a larger package of internet- and intranet-related programs that are used for sending and receiving emails downloading requests for File Transfer Protocol (FTP) files; and building and publishing web pages Many basic web servers will also support server-side scripting, which is used to employ script web server that can customize the response to the client.

Server-side scripting runs on the server machine and typically has a broad feature set, which includes database access. The server-side scripting process will also use Active Server Pages (ASP), Hypertext Preprocessor (PHP) and other scripting languages. This process also allows HTML documents to be created dynamically.

1.3 Dynamic vs. static web servers

A web server can be used to serve either static or dynamic content. Static refers to the content being shown as is, while dynamic content can be updated and changed. A static web server will consist of a computer and HTTP software. It is considered static because the sever will send hosted files as is to a browser.

Dynamic web browsers will consist of a web server and other software such as an application server and database. It is considered dynamic because the application server can be used to update any hosted files before they are sent to a browser. The web server can generate content when it is requested from the database. Though this process is more flexible, it is also more complicated.

1.4 Common web server are as follows

There are a number of common web servers available, some including

Apache HTTP Server. Developed by Apache Software Foundation, it is a free and open source web server for Windows, Mac OS X, Unix, Linux, Solaris and other operating systems; it needs the Apache license.

Microsoft Internet Information Services (IIS). Developed by Microsoft for Microsoft platforms; it is not open sourced, but widely used.

Nginx. A popular open source web server for administrators because of its light resource utilization and scalability. It can handle many concurrent sessions due to its event-driven architecture. Nginx also can be used as a proxy server and load balancer.

Lighttpd. A free web server that comes with the FreeBSD operating system. It is seen as fast and secure, while consuming less CPU power.

Sun Java System Web Server. A free web server from Sun Microsystems that can run on Windows, Linux and Unix. It is well-equipped to handle medium to large websites.

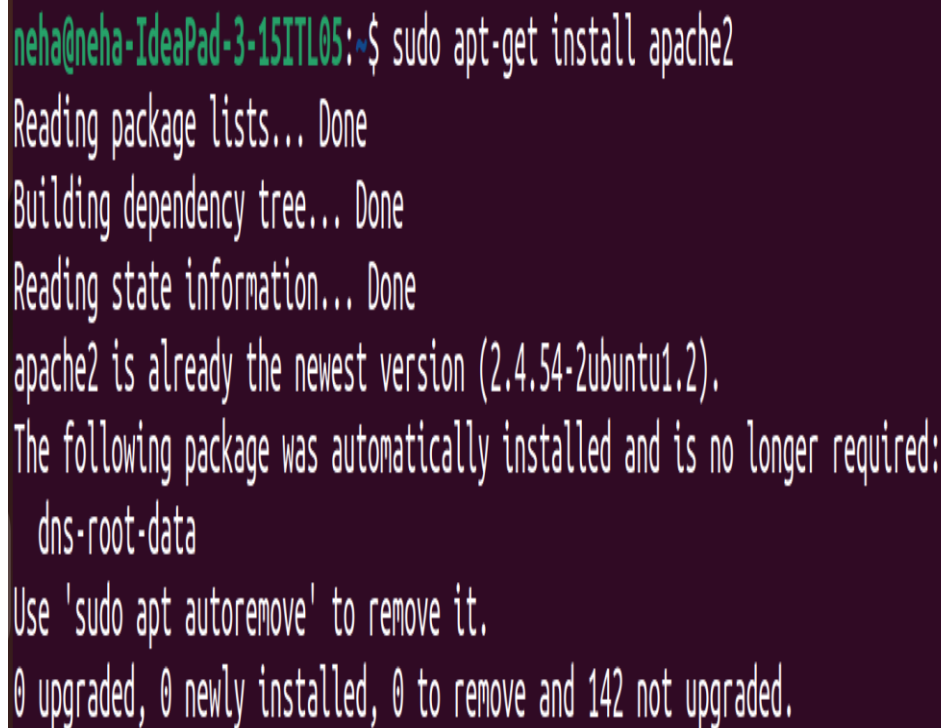
HTTPD - APACHE2 WEB SERVER

Apache is the most commonly used Web server on Linux systems. Web servers are used to serve Web pages requested by client computers. Clients typically request (FTP) file transfer protocol the most common protocol used to transfer the web pages, a protocol for uploading and downloading files, are also supported. and view Web pages using Web browser applications such as Firefox, Opera, Chromium, or Internet Explorer.

2.1 To install Apache2

At a terminal prompt enter the following command

```
$ sudo apt-get install apache2
```



```
neha@neha-IdeaPad-3-15ITL05:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.54-2ubuntu1.2).
The following package was automatically installed and is no longer required:
  dns-root-data
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 142 not upgraded.
```

fig 2.1

After installation to access the web page of apache2 server open the firefox then type local host the default page is visible as following as shown in fig2.12

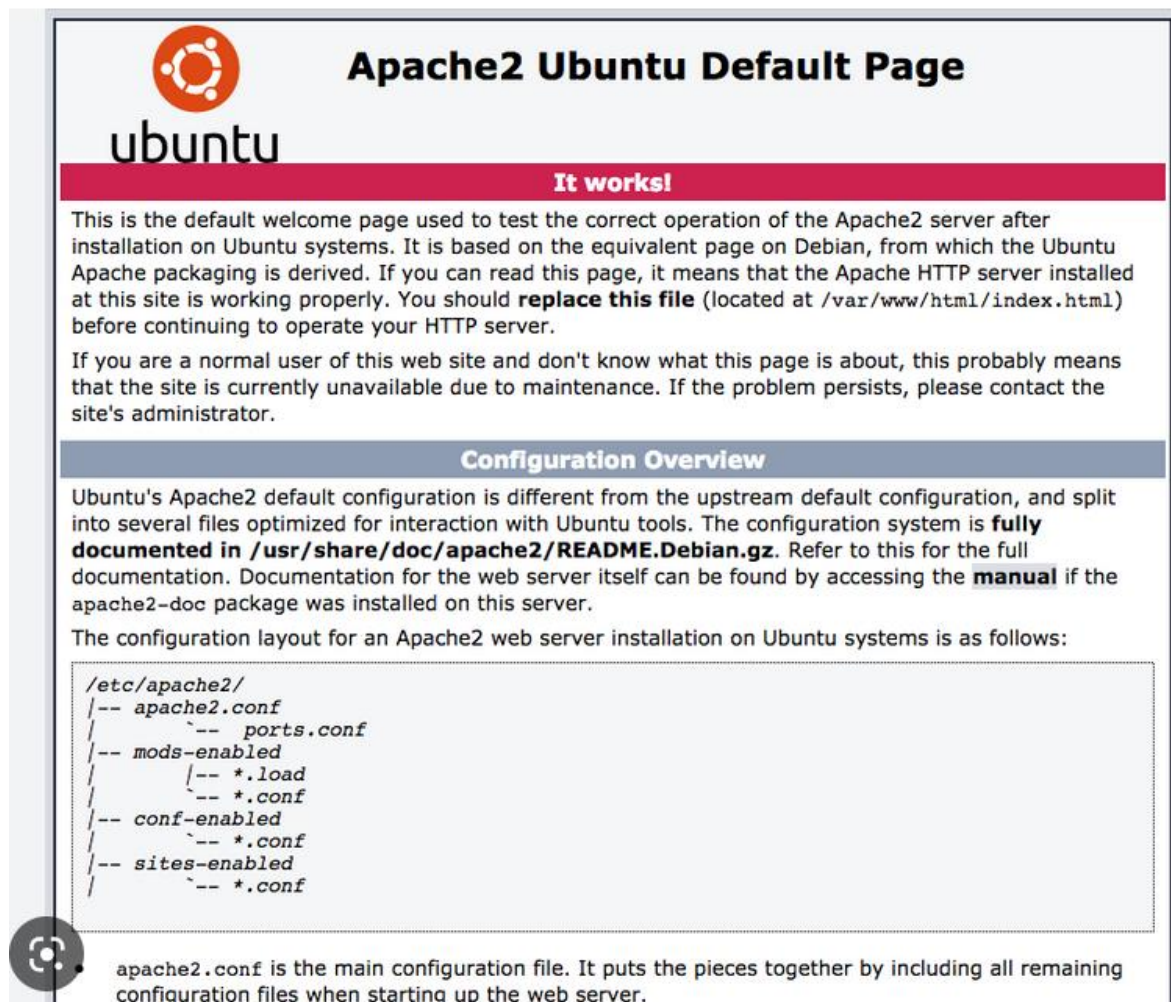


fig 2.12

2.2 Apache2 is configured by placing *directives* in plain text configuration files. These *directives* are separated between the following files and directories

apache2.conf the main Apache2 configuration file. Contains settings that are *global* to Apache2. *httpd.conf* historically the main Apache2 configuration file, named after the httpd daemon. In other distributions (or older versions of Ubuntu), the file might be present. In Ubuntu, all configuration options have been moved to *apache2.conf* and the below referenced directories, and this file no longer exists.

conf-available this directory contains available configuration files. All files that were previously in `/etc/apache2/conf.d` should be moved to `/etc/apache2/conf-available`.

E325: ATTENTION

Found a swap file by the name "/var/www/html/.index.html.swp"

owned by: root dated: Thu Mar 30 16:35:05 2023

file name: /var/www/html/index.html

modified: YES

user name: root host name: neha-IdeaPad-3-15ITL05

process ID: 8668

While opening file "/var/www/html/index.html"

dated: Thu Apr 20 19:18:32 2023

NEWER than swap file!

(1) Another program may be editing the same file. If this is the case, be careful not to end up with two different instances of the same file when making changes. Quit, or continue with caution.

(2) An edit session for this file crashed.

If this is the case, use ":recover" or "vim -r /var/www/html/index.html" to recover the changes (see ":help recovery").

If you did this already, delete the swap file "/var/www/html/.index.html.swp" to avoid this message.

Swap file "/var/www/html/.index.html.swp" already exists!

[O]pen Read-Only, (E)dit anyway, (R)ecover, (D)elete it, (Q)uit, (A)bort: █

fig 2.21

conf-enabled holds *symlinks* to the files in /etc/apache2/conf-available. When a configuration file is symlinked, it will be enabled the next time apache2 is restarted.*envvars* file where Apache2 *environment* variables are set.

mods-available this directory contains configuration files to both load *modules* and configure them. Not all modules will have specific configuration files, however.*mods-enabled* holds *symlinks* to the files in /etc/apache2/mods-available. When a module configuration file is sym linked it will be enabled the next time apache2 is restarted.

Ports.conf houses the directives that determine which TCP ports Apache2 is listening on.

sites-available this directory has configuration files for Apache2 *Virtual Hosts*. Virtual Hosts allow Apache2 to be configured for multiple sites that have separate configurations.

sites-enabled like *mods-enabled*, *sites-enabled* contains symlinks to the `/etc/apache2/sites-`

```
<html>
  <head></head>
  <body>
    <center><b>hi welcome to our apache2 server</b></center>
  </body>
</html>
```

4,15-29 All

fig 2.2

available directory. Similarly when a configuration file in sites-available is symlinked, the site configured by it will be active once Apache2 is restarted.

magic instructions for determining MIME type based on the first few bytes of a file.

In addition, other configuration files may be added using the *Include* directive, and wildcards can be used to include many configuration files. Any directive may be placed in any of these configuration files. Changes to the main configuration files are only recognized by Apache2 when it is started or restarted.

The server also reads a file containing mime document types; the filename is set by the *TypesConfig* directive, typically via `/etc/apache2/mods-available/mime.conf`, which might also include additions and overrides, and is `/etc/mime.types` by default.

2.3 Basic Settings

Apache2 ships with a virtual-host-friendly default configuration. That is, it is configured with a single default virtual host (using the *VirtualHost* directive) which can be modified or used as-is if you have a single site, or used as a template for additional virtual hosts if you have multiple sites. If left alone, the default virtual host will serve as your default site, or the site users will see if the URL they enter does not match the *ServerName* directive of any of your custom sites. To modify the default virtual host, edit the file `/etc/apache2/sites-available/000-default.conf`.

Fig 2.31

To configure a new virtual host or site, copy that file into the same directory with a name you choose. For example

```
$sudo /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/mynewsite.conf
```

to access the web page of apache2 server open the firefox then type local host the default page is visible as following as shown in fig 2.32

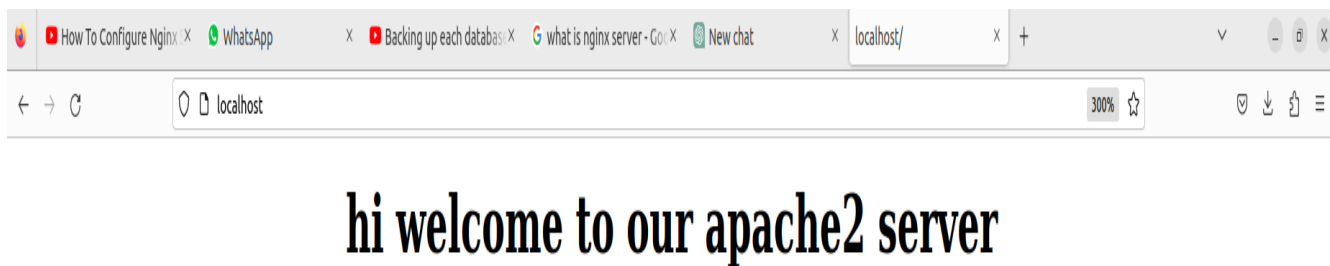


fig 2.32

WHAT IS NGINX?

NGINX is pronounced as "engine-ex". It is an open-source, fast, lightweight and high-performance web server that can be used to serve static files. NGINX has considered as the popular web server behind the Apache web server and Microsoft's IIS. In its initial release, NGINX functioned for HTTP web serving. Today, however, it also serves as a reverse proxy server for HTTP, HTTPS, SMTP, IMAP, POP3 protocols, on the other hand, it is also used for HTTP load balancer, HTTP cache, and email proxy for IMAP, POP3, and SMTP. NGINX improves content and application delivery, improves security, and facilitates scalability and availability for the busiest websites on the internet. In short, we can say that Nginx is just a kind of software that is used in web servers to serve concurrent requests. Previously we used to install Apache in web servers to handle these functions, but as the world, in growing and demanding more things at one time, the term concurrency comes into the world and nginx launched for the same thing.

3.1 To install Nginx on Ubuntu, follow these steps

Open a terminal window by pressing Ctrl+Alt+T or by searching for "Terminal" in the applications menu.

Update the package list by running the following command

\$ sudo apt-get update

Install Nginx by running the following command

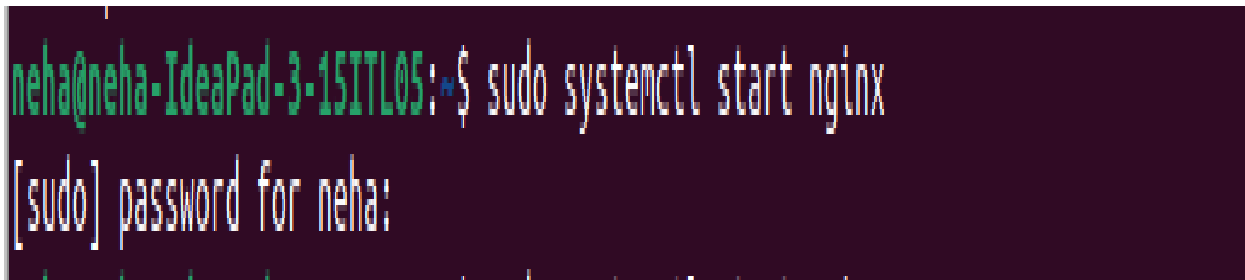
\$ sudo apt-get install nginx

```
neha@neha-IdeaPad-3-15ITL05:~$ sudo apt-get install nginx
[sudo] password for neha:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  dns-root-data
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx-common nginx-core
Suggested packages:
  fcgiwrap nginx-doc
The following NEW packages will be installed:
  libnginx-mod-http-geoip2 libnginx-mod-http-image-filter
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip2 nginx nginx-common nginx-core
0 upgraded, 9 newly installed, 0 to remove and 142 not upgraded.
Need to get 725 kB of archives.
After this operation, 2,450 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 nginx-common all 1.22.0-1ubuntu1.1 [43.3 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 libnginx-mod-http-geoip2 amd64 1.22.0-1ubuntu1.1 [14.7 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 libnginx-mod-http-image-filter amd64 1.22.0-1ubuntu1.1 [18.1 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.22.0-1ubuntu1.1 [16.4 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 libnginx-mod-mail amd64 1.22.0-1ubuntu1.1 [50.3 kB]
Get:6 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 libnginx-mod-stream amd64 1.22.0-1ubuntu1.1 [75.9 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu kinetic-updates/main amd64 libnginx-mod-stream-geoip2 amd64 1.22.0-1ubuntu1.1 [1
```

fig 3.1

After the installation is complete, start the Nginx service by running the following command

\$ sudo systemctl start nginx

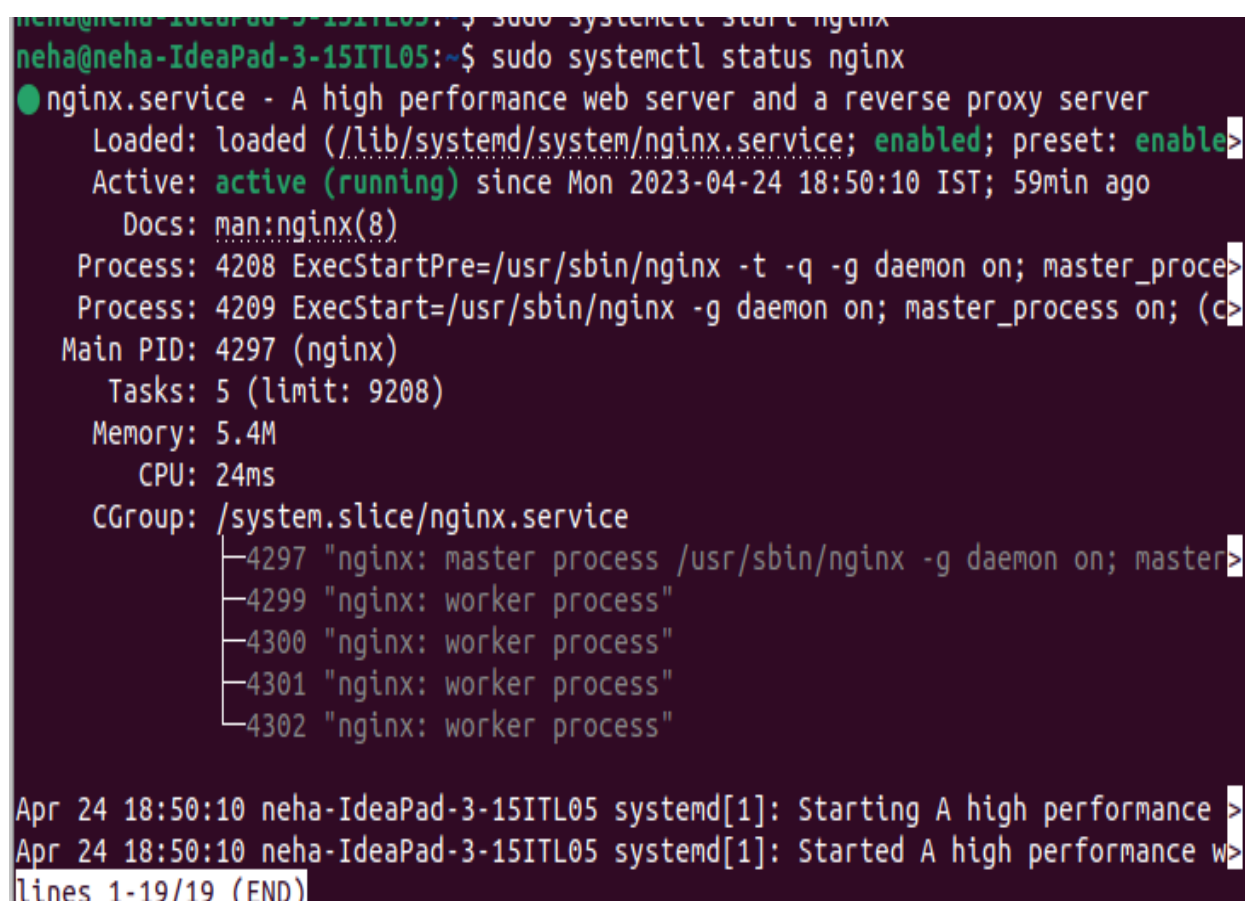


```
neha@neha-IdeaPad-3-15ITL05:~$ sudo systemctl start nginx
[sudo] password for neha:
```

fig 3.2

You can check the status of the Nginx service by running the following command

\$ sudo systemctl status nginx



```
neha@neha-IdeaPad-3-15ITL05:~$ sudo systemctl status nginx
neha@neha-IdeaPad-3-15ITL05:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enable>
   Active: active (running) since Mon 2023-04-24 18:50:10 IST; 59min ago
     Docs: man:nginx(8)
  Process: 4208 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_proce>
  Process: 4209 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (c>
 Main PID: 4297 (nginx)
    Tasks: 5 (limit: 9208)
   Memory: 5.4M
      CPU: 24ms
   CGroup: /system.slice/nginx.service
           └─4297 "nginx: master process /usr/sbin/nginx -g daemon on; master>
             └─4299 "nginx: worker process"
               └─4300 "nginx: worker process"
                 └─4301 "nginx: worker process"
                   └─4302 "nginx: worker process"

Apr 24 18:50:10 neha-IdeaPad-3-15ITL05 systemd[1]: Starting A high performance >
Apr 24 18:50:10 neha-IdeaPad-3-15ITL05 systemd[1]: Started A high performance w>
lines 1-19/19 (END)
```

fig3.3

After installation the default web page is shown in fig 3.4



PROXY SERVER

In computer networking, a **proxy server** is a server application that acts as an intermediary between a client requesting a resources and the server providing that resource.

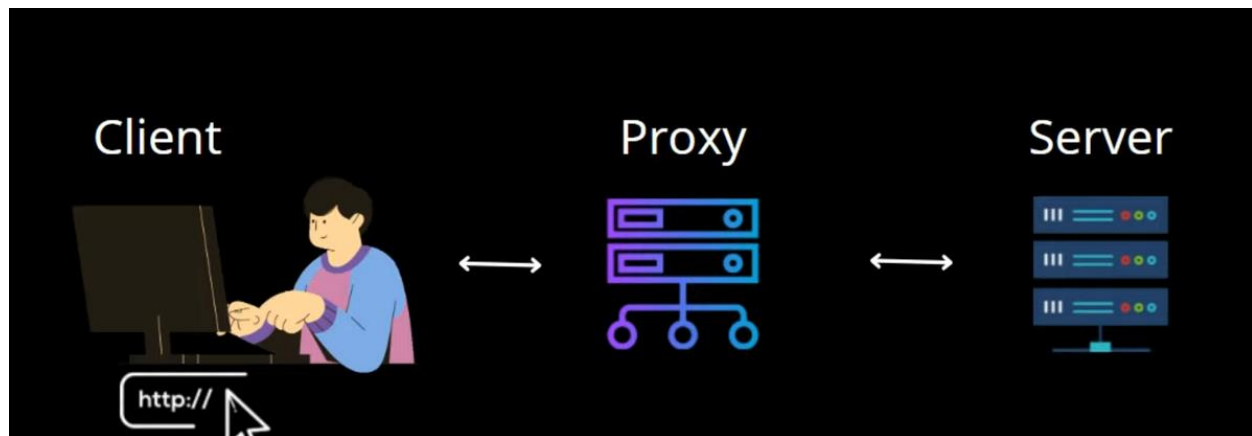


Fig 4.1

Instead of connecting directly to a server that can fulfill a request for a resource, such as a file or web page, the client directs the request to the proxy server, which evaluates the request and performs the required network transactions. This serves as a method to simplify or control the complexity of the request, or provide additional benefits such as load balancing, privacy, or security. Proxies were devised to add structure and encapsulation on distributed systems. A proxy server thus functions on behalf of the client when requesting service, potentially masking the true origin of the request to the resource server.

4.1 Types of proxy server

A proxy server may reside on the user's local computer, or at any point between the user's computer and destination servers on the internet. A proxy server that passes unmodified requests and responses is usually called a gateway or sometimes a *tunneling proxy*. A forward proxy is an Internet-facing proxy used to retrieve data from a wide range of sources (in most cases anywhere on the Internet). A reverse proxy is usually an internal-facing proxy used as a front-end to control and protect access to a server on a private network. A reverse proxy commonly also performs tasks such as load balancing authentication, description and caching

4.2 What are forward proxies and reverse proxies?

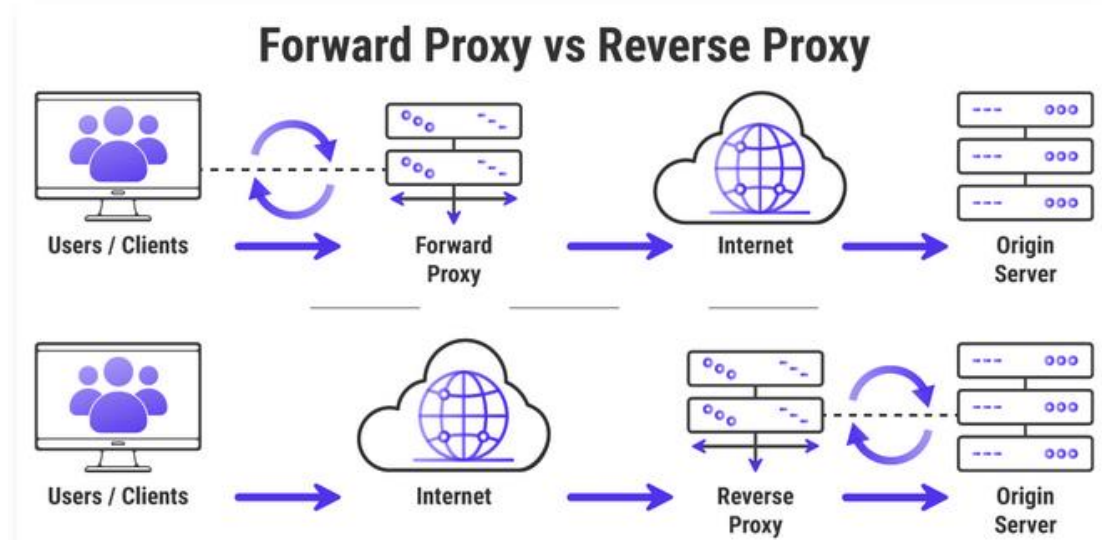


Fig 4.2

The forward proxy accepts connections from computers on a private network and forwards those requests to the public internet. It is the single point of exit for subnet users who want to access resources outside of their private network.

As the name implies, a reverse proxy is the opposite of a forward proxy. The reverse proxy acts as a single point of entry for external systems to access resources on a private subnet.

In an enterprise architecture, a reverse proxy acts as the public access point for users to access data and information that is stored on servers that reside in a private, isolated subnet.

4.3 Why Use a Reverse Proxy?

Many businesses, especially large enterprises, use bespoke websites that are tailor-made to their unique needs and aren't running on WordPress. Some examples include bank and insurance websites. In other cases, a business may host their site on an external service that doesn't allow them to install any external software (e.g. wordpress, these are small to mid-sized retailers using an e-commerce platform). Since WordPress has robust cms features, many businesses, including large enterprises with bespoke websites, may prefer to host their blogs using wordpress.

One way to get around this problem is to install wordpress on the main website's subdomain and structure the navigation menus such that users can switch easily between the main website and the

blog. Since subdomains behave as a unique domain, it can affect your sites SEO. Even though Google treats both subdomains and subdirectories equally, it takes more effort to optimize a website for search engine rankings if it's hosted on a subdomain if than its is hosted in a sub directory.

4.4 nginx reverse proxy setup for apache2

The nginx servers as reverse proxy server as well as act as a load balancer using these command Configure Nginx to Proxy Requests Next, you need to configure Nginx to proxy requests for domains hosted on Apache. To do that, create a new virtual host file. here using the **nano** editor to add the code, but you can use any code editor.

\$cd /etc/nginx/sites-available

```
##

ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
ssl_prefer_server_ciphers on;

##
# Logging Settings
##

access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

##
# Gzip Settings
##

gzip on;

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript te

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
```

\$ sudo nano default


```
#mail {  
#     # See sample authentication script at:  
#     # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript  
#  
#     # auth_http localhost/auth.php;  
#     # pop3_capabilities "TOP" "USER";  
#     # imap_capabilities "IMAP4rev1" "UIDPLUS";  
#  
#     server {  
#         listen    localhost:110;  
#         protocol  pop3;  
#         proxy     on;  
#     }  
#  
#     server {  
#         listen    localhost:143;  
#         protocol  imap;  
#         proxy     on;  
#     }  
#}
```

fig 4.2

write the code as follows

```
#default configuration as follows  
upstream samplecluster {  
    server localhost110;  
    server localhost143; }  
server {  
    listen 80;  
    default server;  
    listen[]80 default_server; }
```

Save the Virtual Host File Created Then activate the new virtual host by creating a symlink for the files named **example.com.conf** in both the **/etc/nginx/sites-available** and the **/etc/nginx/sites-enabled** directories.

Test Nginx for Errors

After that, test Nginx for any configuration errors.

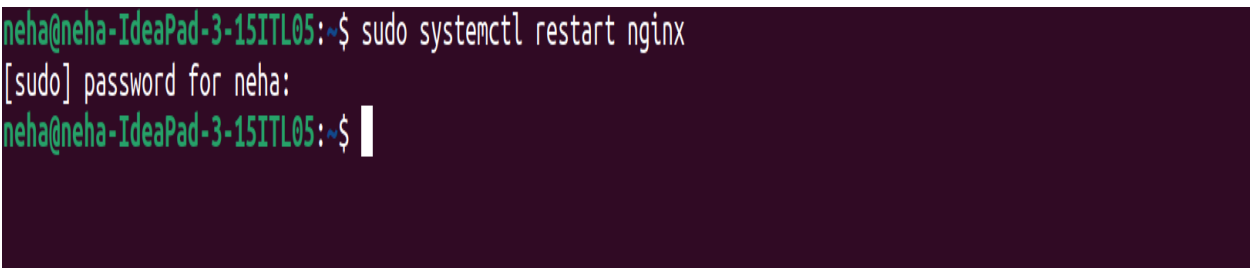
```
sudo nginx -t
```

If there are no errors, reload Nginx to enforce the changes.

```
sudo systemctl reload nginx
```

then restart the nginx server using this command

```
$ sudo systemctl restart nginx
```



```
neha@neha-IdeaPad-3-15ITL05:~$ sudo systemctl restart nginx
[sudo] password for neha:
neha@neha-IdeaPad-3-15ITL05:~$
```

fig 4.3

then check the status of the of nginx server using this command

```
$sudo systemctl status nginx
```

```
neha@neha-IdeaPad-3-15ITL05:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Mon 2023-04-24 18:50:10 IST; 59min ago
     Docs: man:nginx(8)
  Process: 4208 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on;
  Process: 4209 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=0)
 Main PID: 4297 (nginx)
    Tasks: 5 (limit: 9208)
   Memory: 5.4M
      CPU: 24ms
   CGroup: /system.slice/nginx.service
           └─4297 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─4299 "nginx: worker process"
               └─4300 "nginx: worker process"
                 └─4301 "nginx: worker process"
                   └─4302 "nginx: worker process"

Apr 24 18:50:10 neha-IdeaPad-3-15ITL05 systemd[1]: Starting A high performance web server: nginx.
Apr 24 18:50:10 neha-IdeaPad-3-15ITL05 systemd[1]: Started A high performance web server: nginx.
lines 1-19/19 (END)
```

4.5 Benefits of Using a Reverse Proxy

Besides the above use case, reverse proxies also grant many other benefits. The section below discusses some of their major advantages

LOAD BALANCING

A single origin server cannot handle all the incoming traffic for a website with millions of daily unique visitors. In these cases, you can distribute the traffic smartly among a pool of many servers.

Usually, all the servers will host the same content to eliminate a single point of failure, making the website more reliable. A reverse proxy is a great way to set this up as it can receive the incoming traffic before it reaches the origin server. If the origin server is overloaded or fails completely, it can distribute the traffic to other servers without affecting the site functionality. Reverse proxies can also direct the incoming requests to several servers, with each server performing a specific function it's optimized for. The reverse proxy can then gather responses from all the servers and deliver them to the client. Since we use most of the popular reverse proxies primarily for load balancing, they're also referred to as **Load Balancers**.

Global Server Load Balancing (GSLB)

GSLB is an advanced load balancing method for distributing website traffic among many servers placed strategically around the world. It's typically done via any cast routing techniques, where the reverse proxy picks the server node based on the fastest travel time between the client and the server. Not only does GSLB increase the site's reliability and security considerably, it also reduces latency and load times, thereby enhancing user experience. You can use GSLB with other network optimization techniques such as spoon feeding to free up the origin servers' computational resources even more. Though you can set up Global Server Load Balancing manually on your server, it's usually taken care of by dedicated CDNs such as cloud fare and key CDN (which also powers [kinsta](#) CDN).

Enhanced Security

Reverse proxies can cloak the IP address and other characteristics of origin servers. Thus, your website's origin server can maintain its anonymity better, increasing its security significantly. Since the reverse proxy will receive all the traffic before it reaches the main server, any attackers hackers will find it harder to target your website with security threats such as D Dos attacks. You can use a strict firewall to harden the reverse proxy with tighter security against common cyber-attacks. Without a reverse proxy installed, it is difficult to remove malware or start take downs.

CONCLUSION

A proxy server is a computer or an application that acts as an intermediary between a client and a server. It facilitates communication between these two entities by forwarding requests from clients to servers and returning responses back to the clients. Proxy servers offer several benefits and serve various purposes, leading to the following conclusion

Enhanced Privacy and Security Proxy servers can improve privacy and security by acting as a buffer between clients and servers. They can hide the client's IP address, making it difficult for servers to identify the origin of the request. This feature is especially useful in maintaining anonymity and protecting sensitive information.

Content Filtering and Access Control Proxy servers can be used to filter and control access to websites and content. Organizations often employ proxy servers to restrict access to certain websites or types of content, helping enforce internet usage policies and prevent unauthorized access.

Caching and Performance Optimization Proxy servers can cache frequently accessed resources, such as web pages or media files. This caching mechanism improves performance by serving cached content directly to clients, reducing the load on the original server and decreasing response times for subsequent requests.

Bandwidth Optimization Proxy servers can compress and optimize data, reducing the amount of bandwidth required for transferring content between clients and servers. This feature is particularly beneficial in low-bandwidth environments or when dealing with limited data plans.

Load Balancing Proxy servers can distribute incoming requests across multiple servers, balancing the load and ensuring optimal utilization of available resources. This technique improves the overall performance and scalability of the server infrastructure.

BIBLIOGRAPHY

1. Cherita L. Corbett, Raheem A. Beyah, John A. Copeland, Using Active Scanning to Identify Wireless NICs, in Proceedings of the 7th IEEE Workshop on Information Assurance, U.S. Military Academy, West Point, NY, 21-23 June 2006.
2. Pranab Kumar Chakravarty, Computer Networking Technologies and Application to IT Enabled Services.
3. Antonio Carzaniga, Basic concepts in Computer Networking, September 19, 2014.
4. TeodoraBakardjieva, Introduction to Computer Networking.
5. Peter L. Dordal, An Introduction to Computer Networks, Release 1.8.07, June 16, 2015.
6. Bob Dickerson, Computer Networks, January 2005.
7. Russell Anthony Tantillo, Network Security through Open Source Intrusion Detection Systems, May 2012.