Apache HTTP Server (httpd):

What is Apache HTTP Server?

- Apache HTTP Server, often referred to as Apache, is an open-source web server software maintained by the Apache Software Foundation. It is one of the most widely used web servers globally.

Key Features:

- Open Source: Apache is open-source, which means its source code is freely available for modification and distribution.
- Modular Architecture: Apache's modular architecture allows for the easy addition of features and functionality through modules.
- Cross-Platform: It is designed to run on various operating systems, including Linux, Unix, Windows, and more.
- Robust Security: Apache provides numerous security features, including SSL/TLS support and authentication modules.

Components:

- httpd:This is the main Apache server executable.
- Configuration Files: Apache's configuration files are typically found in `/etc/httpd/` or `/etc/apache2/` directory on Linux systems.

Configuration:

- The main configuration file is often named `httpd.conf`. However, Apache's configuration is split across multiple files, and additional configurations can be included from separate files.

Modules:

- Apache uses modules to extend its functionality. Common modules include mod_ssl for SSL/TLS support, mod_rewrite for URL rewriting, and mod_proxy for proxy capabilities.

Virtual Hosts:

- Apache supports virtual hosts, allowing you to host multiple websites on a single server. Each virtual host has its configuration.

Logging:

- Apache generates logs that record various aspects of its operation, including access logs (access to resources) and error logs (errors encountered during operation).

Security:

- Apache provides various security features, including authentication modules (like mod_auth_basic) and access control directives.

Restarting/Stopping/Starting Apache:

- To restart Apache on Linux, you can use the following command:

```
sudo systemctl restart apache2  # For systems using systemd
```

Common Directories:

- `/var/www/html`: Default root directory for the Apache web server.
- `/etc/httpd/` or `/etc/apache2/`: Configuration files directory.

Package Managers:

- On Debian/Ubuntu, you use `apt`, and on Red Hat-based systems, you use `yum` or `dnf` to install Apache.

Practice:

Get it installed in you machine first

A=apt-get install apache2

Cd /etc/apache2

At this path you will see all the files/directories related to apache server

Open the apache2.conf file to see the details

Less apache2.conf

Now you can check the default html page getting hosted on the server

Cd /var/www/html/index.html

You can edit this default page

vim index.html

You can check the status of apache2

service apache2 status

Or

systemctl status apache2

To start the server

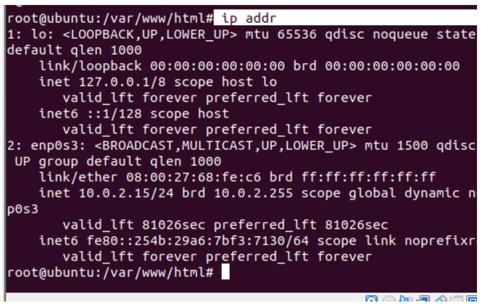
systemctl start apache2

To stop the server

systemctl stop spache2

Check the IP address of your VM

ip addr



```
root@ubuntu:/var/www/html# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
 UP group default qlen 1000
    link/ether 08:00:27:68:fe:c6 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic n
p0s3
       valid_lft 81026sec preferred_lft 81026sec
    inet6 fe80::254b:29a6:7bf3:7130/64 scope link noprefixr
       valid_lft forever preferred_lft forever
root@ubuntu:/var/www/html#
```

Now check with this Ip if you are able to see that static page of Apache but incase if its not working then the firewall settings of Linux stopping it

You can stop the firewall for a while just to check it then start the firewall server again

We will use the highly restrictive profile ‘Apache’ to enable network activity on port 80.

Firewall configuration

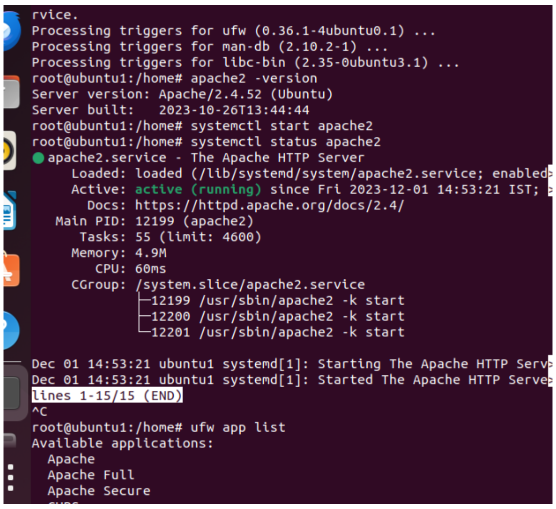
apt-get install ufw

ufw allow ‘Apache’

ufw status

Configuring Apache web server. Verifying Apache service

systemctl status apache2

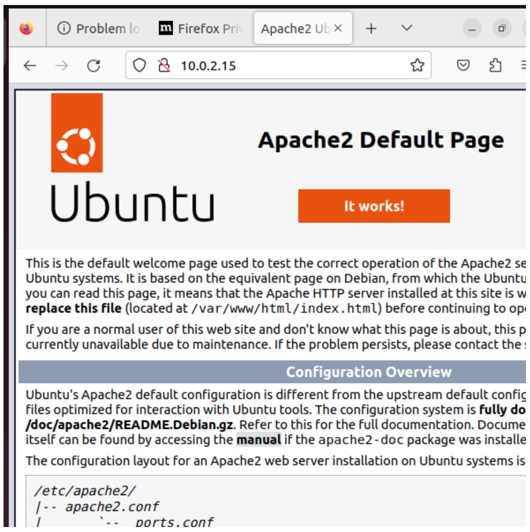


```
rvice.
Processing triggers for ufw (0.36.1-4ubuntu0.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
root@ubuntu1:/home# apache2 -version
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2023-10-26T13:44:44
root@ubuntu1:/home# systemctl start apache2
root@ubuntu1:/home# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled
     Active: active (running) since Fri 2023-12-01 14:53:21 IST;
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 12199 (apache2)
      Tasks: 55 (limit: 4600)
     Memory: 4.9M
        CPU: 60ms
     CGroup: /system.slice/apache2.service
             ├─12199 /usr/sbin/apache2 -k start
             ├─12200 /usr/sbin/apache2 -k start
             └─12201 /usr/sbin/apache2 -k start

Dec 01 14:53:21 ubuntu1 systemd[1]: Starting The Apache HTTP Serv
Dec 01 14:53:21 ubuntu1 systemd[1]: Started The Apache HTTP Serve
lines 1-15/15 (END)
^C
root@ubuntu1:/home# ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
```

Another approach to verify if Apache is running fine by requesting a web page from the Apache web server. To do so, find your IP address using the following command:

Then open the web browser and access Apache welcome page as follows:

hostname –I



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 se
Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu
you can read this page, it means that the Apache HTTP server installed at this site is w
replace this file (located at `/var/www/html/index.html`) before continuing to op

If you are a normal user of this web site and don't know what this page is about, this p
currently unavailable due to maintenance. If the problem persists, please contact the

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default confic
files optimized for interaction with Ubuntu tools. The configuration system is **fully do**
/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Docume
itself can be found by accessing the **manual** if the apache2-doc package was installe

The configuration layout for an Apache2 web server installation on Ubuntu systems is

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
```

#NginX Server:

Nginx (pronounced **"engine-x")** is a popular open-source web server, reverse proxy server, and load balancer. It is widely used for serving static content, handling SSL/TLS termination, and efficiently managing concurrent connections. Originally created by Igor Sysoev in 2004, Nginx has gained significant popularity due to its performance, scalability, and flexibility.

Here are some key features and aspects of Nginx:

1. Web Server:

- Nginx is commonly used as a web server to serve static content, such as HTML, CSS, and images. It excels in handling a large number of simultaneous connections with low resource usage.

2. Reverse Proxy:

- Nginx can act as a reverse proxy, forwarding client requests to other servers (e.g., application servers or microservices). This enables Nginx to handle tasks like load balancing and caching.

3. Load Balancer:

- As a load balancer, Nginx can distribute incoming traffic across multiple servers to ensure optimal resource utilization and high availability. This is particularly useful for distributing the load among application servers to prevent overload on any single server.

4. SSL/TLS Termination:

- Nginx can handle SSL/TLS termination, offloading the decryption process from the application servers. This helps in reducing the computational load on backend servers and simplifying the management of SSL certificates.

5. High Performance:

- Nginx is designed for high performance and efficiency. It is known for its low memory usage and ability to handle a large number of concurrent connections, making it suitable for serving static content and acting as a reverse proxy.

6. **Configuration**:

- Nginx uses a declarative configuration syntax that is easy to understand and maintain. The configuration files are typically located in the `/etc/nginx` directory, and changes can be applied without restarting the server, making it convenient for dynamic environments.

7. Modules:

- Nginx supports a modular architecture, allowing users to extend its functionality through various modules. There are modules available for features like caching, security, and more.

8. Community and Support:

- Nginx has a large and active community, providing extensive documentation, tutorials, and support. It is widely used by websites, applications, and organizations of all sizes.

Practice:

Before installing any software it's a good idea to update your apt (short for Ubuntu's default package manager aptitude) software list:

apt-get update

Install nginx server

apt-get install nginx

If not installed already then first install ufw in ubuntu machine

We suppose **ufw** is enabled on your Ubuntu 20.04 system. You may double check its status:

apt-get install ufw

ufw status

ufw app list → with this you can see the rules in firewall which are already enabled

ufw allow 'Nginx HTTP' → Add the rule for nginx server

Let's view the status of our nginx service by typing

systemctl status nginx

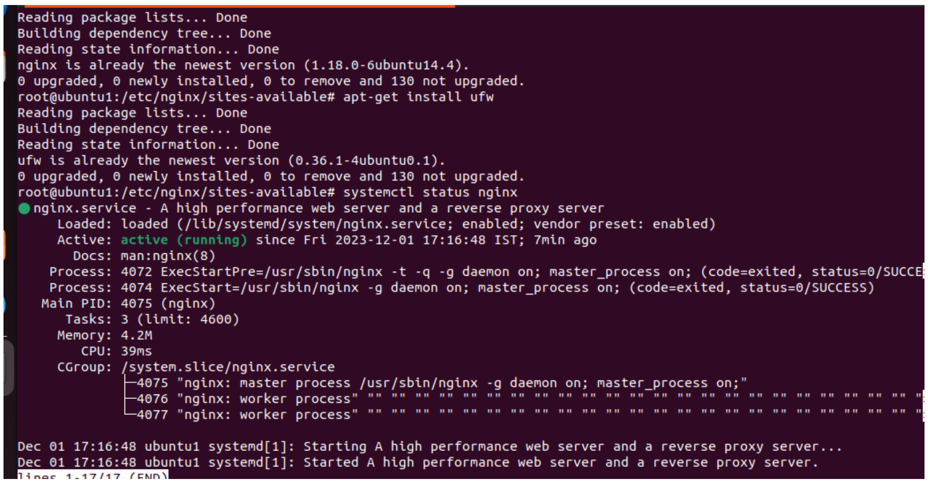
systemctl start nginx

systemctl restart nginx

systemctl reload nginx

Should we want to stop nginx service until the next system reboot, we will use :

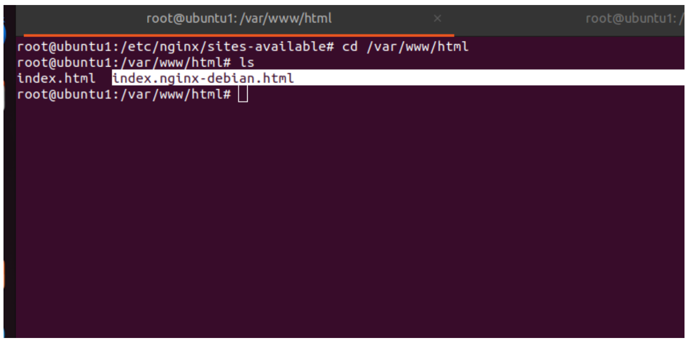
systemctl stop nginx



```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.18.0-6ubuntu14.4).
0 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
root@ubuntu1:/etc/nginx/sites-available# apt-get install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.1-4ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
root@ubuntu1:/etc/nginx/sites-available# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2023-12-01 17:16:48 IST; 7min ago
       Docs: man:nginx(8)
    Process: 4072 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCE
    Process: 4074 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 4075 (nginx)
      Tasks: 3 (limit: 4600)
     Memory: 4.2M
        CPU: 39ms
     CGroup: /system.slice/nginx.service
             ├─4075 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─4076 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "
             └─4077 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "

Dec 01 17:16:48 ubuntu1 systemd[1]: Starting A high performance web server and a reverse proxy server...
Dec 01 17:16:48 ubuntu1 systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-17/17 (END)
```

Nginx serves its content from the /var/www/html directory by default. Let's access this directory by using the Linux cd (change directory) command and list its content with the ls (list directory contents) command.

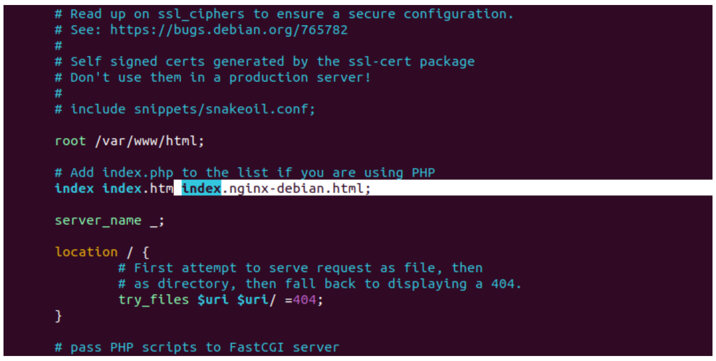
We see a file named **index.nginx-debian.html**



```
root@ubuntu1:/etc/nginx/sites-available# cd /var/www/html
root@ubuntu1:/var/www/html# ls
index.html  index.nginx-debian.html
root@ubuntu1:/var/www/html# 
```

cd /etc/nginx/sites-available/Default

Content of Default file

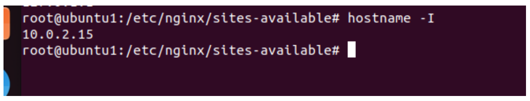


```
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.html index.nginx-debian.html;

        server_name _;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }

        # pass PHP scripts to FastCGI server
```

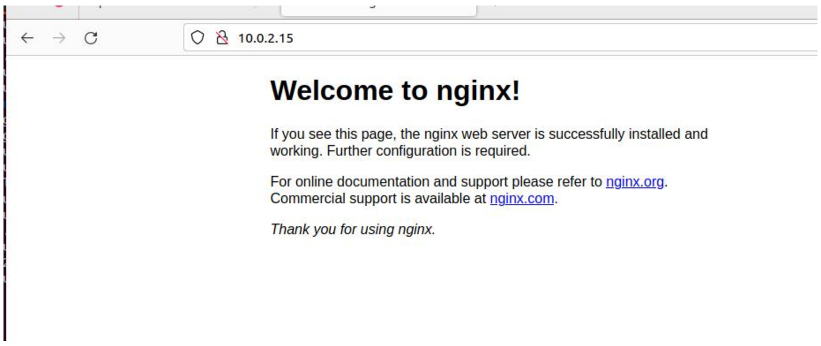
Check the IP of your machine

Hostname -I

And put this IP on browser you will be able to see the default page of nginx



```
root@ubuntu1:/etc/nginx/sites-available# hostname -I
10.0.2.15
root@ubuntu1:/etc/nginx/sites-available# 
```



10.0.2.15

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

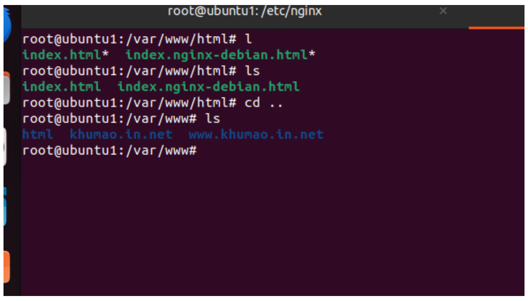
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Configure NGINX Server Blocks to Host Multiple Websites

NGINX can host more than one domain on the same web server by using server blocks. We make a new server block by creating a new directory for our content in /var/www/.

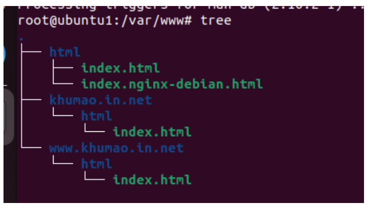
Let’s **make two new directories** for our domains www.khumao.in.net and khumao.in.net in the **/var/www/** directory by using the mkdir (make directory) command with -p flag (creates parent directories).



```
root@ubuntu1: /etc/nginx
root@ubuntu1:/var/www/html# l
index.html*  index.nginx-debian.html*
root@ubuntu1:/var/www/html# ls
index.html  index.nginx-debian.html
root@ubuntu1:/var/www/html# cd ..
root@ubuntu1:/var/www# ls
html  khumao.in.net  www.khumao.in.net
root@ubuntu1:/var/www#
```

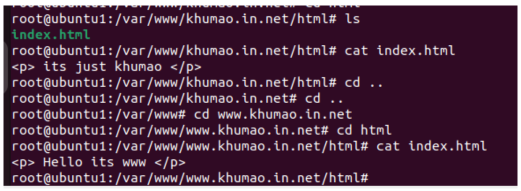
apt-get install tree

For testing purposes let’s create an example **index.html** file in the respective **/var/www/domain_name/html/** directories like this:



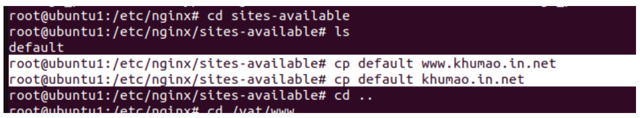
```
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu1:/var/www# tree
.
├── html
│   ├── index.html
│   └── index.nginx-debian.html
├── khumao.in.net
│   └── html
│       └── index.html
└── www.khumao.in.net
    └── html
        └── index.html
```

we put different text messages into the index.html files



```
root@ubuntu1:/var/www/khumao.in.net# cd html
root@ubuntu1:/var/www/khumao.in.net/html# ls
index.html
root@ubuntu1:/var/www/khumao.in.net/html# cat index.html
<p> its just khumao </p>
root@ubuntu1:/var/www/khumao.in.net/html# cd ..
root@ubuntu1:/var/www/khumao.in.net# cd ..
root@ubuntu1:/var/www# cd www.khumao.in.net
root@ubuntu1:/var/www/www.khumao.in.net# cd html
root@ubuntu1:/var/www/www.khumao.in.net/html# cat index.html
<p> Hello its www </p>
root@ubuntu1:/var/www/www.khumao.in.net/html#
```

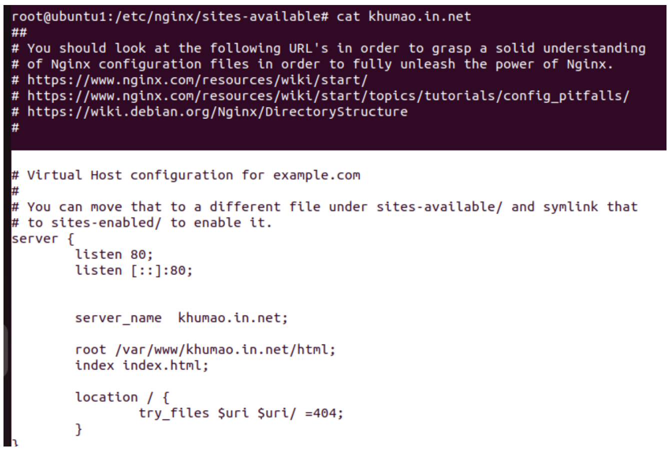
To make the content available we need to **create a server block** and point it to our content. Let’s copy the default configuration file and make two new configuration file templates from it named after our domains: **www.khumao.in.net and khumao.in.net**



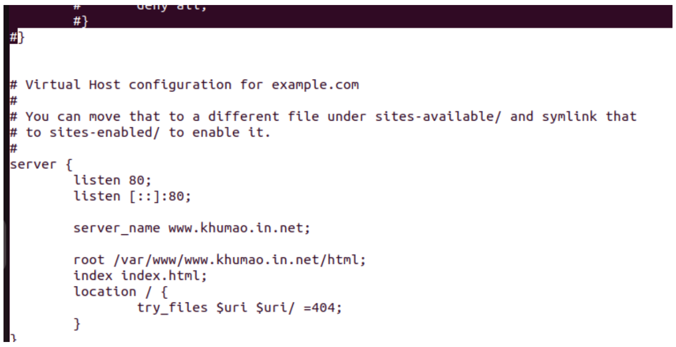
```
root@ubuntu1:/etc/nginx# cd sites-available
root@ubuntu1:/etc/nginx/sites-available# ls
default
root@ubuntu1:/etc/nginx/sites-available# cp default www.khumao.in.net
root@ubuntu1:/etc/nginx/sites-available# cp default khumao.in.net
root@ubuntu1:/etc/nginx/sites-available# cd ..
root@ubuntu1:/etc/nginx# cd /vat/www
```

edit two new configuration files in the /etc/nginx/sites-available/ directory.

Delete “Default server configuration” and un-comment the “Virtual Host configuration […]” section:



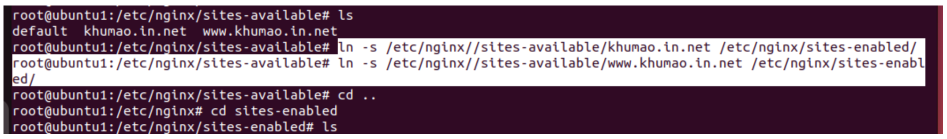
```
root@ubuntu1:/etc/nginx/sites-available# cat khumao.in.net
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
server {
        listen 80;
        listen [::]:80;


        server_name  khumao.in.net;

        root /var/www/khumao.in.net/html;
        index index.html;

        location / {
                try_files $uri $uri/ =404;
        }
}
```



```
        #       deny all;
        #}
#}


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink that
# to sites-enabled/ to enable it.
#
server {
        listen 80;
        listen [::]:80;

        server_name www.khumao.in.net;

        root /var/www/www.khumao.in.net/html;
        index index.html;
        location / {
                try_files $uri $uri/ =404;
        }
}
```

In the server_name line we specify the hostname of our server block, in the root line we specify the folder of our website content, and in the index line we specify what file to serve as the homepage.

To enable our **created server blocks we now need to make symlinks (symbolic links)** of our configuration files and store them in the **/etc/nginx/sites-enabled** directory. In our case we are going to use the following commands:



```
root@ubuntu1:/etc/nginx/sites-available# ls
default  khumao.in.net  www.khumao.in.net
root@ubuntu1:/etc/nginx/sites-available# ln -s /etc/nginx//sites-available/khumao.in.net /etc/nginx/sites-enabled/
root@ubuntu1:/etc/nginx/sites-available# ln -s /etc/nginx//sites-available/www.khumao.in.net /etc/nginx/sites-enabl
ed/
root@ubuntu1:/etc/nginx/sites-available# cd ..
root@ubuntu1:/etc/nginx# cd sites-enabled
root@ubuntu1:/etc/nginx/sites-enabled# ls
```

Change File Ownership and Permissions:

To secure our NGINX server and prevent malicious activity, we need to do a couple more things: change the ownership of our server blocks, and also modify their permissions.

Let’s now change file ownership rights for your hosted website.

chown -R www-data:www-data /var/www/

Here we use **chown** (change owner) command with -R (recursive) option, **www-data:www-data setting for [user]:[group]** (**www-data is the default nginx user on Ubuntu**), and /var/www/ file directory which we use to serve web content and are assigning to www-data user.

Finally, we should change file permissions for your hosted website.

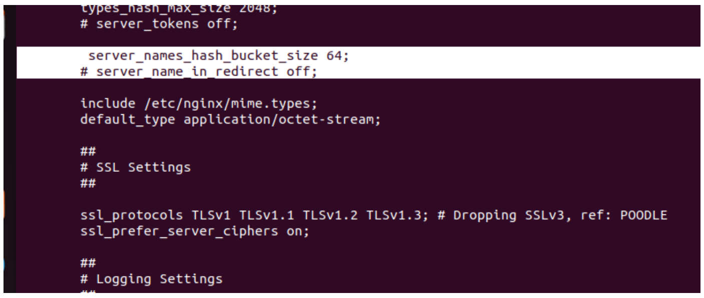
chmod -R 755 /var/www/

Here we use chmod (change file mode) command with -R (recursive) option, 755 permissions (#1(owner) 7 – read, write, execute; #2(group) 5 – read, execute; #3(public) 5 – read, execute) and /var/www/ file directory which permissions we are modifying.

By performing these steps, we only allow the owner (www-data) to modify files and leave other users with read (view web pages and images) and execute (run scripts) permissions.

Also, you should keep www-data user's permissions as low as possible, so that in case your nginx server is breached www-data user couldn't use system utilities, nor access other processes.

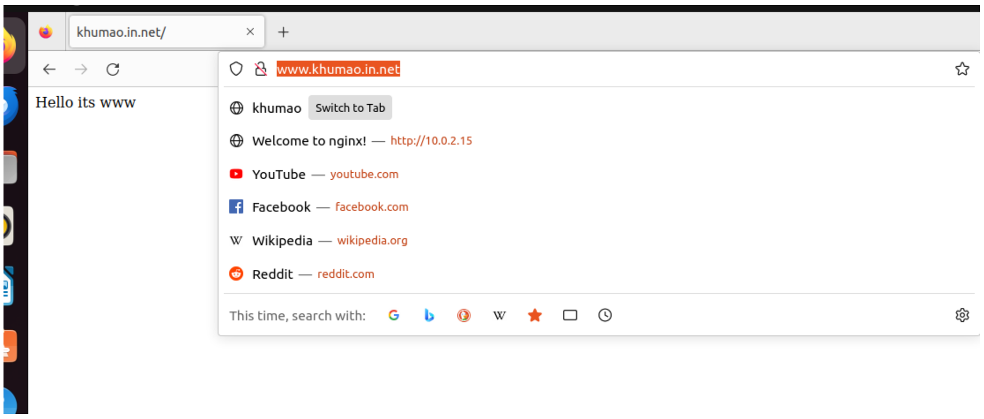
When using multiple server names nginx may run into a hash memory problem. To avoid this, we need to uncomment a single line in the **file /etc/nginx/nginx.conf. The parameter is **server_names_hash_bucket_size 64;**



```
        types_hash_max_size 2048;
        # server_tokens off;

         server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
        ##

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

        ##
        # Logging Settings
```

You may double-check if there are no syntax errors in any of your Nginx configuration files by using the nginx -t command.

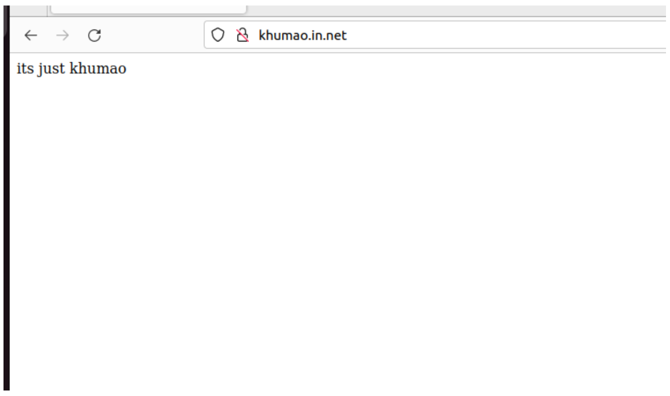
If all is well, restart Nginx with **systemctl restart nginx** for your changes to come into effect.



khumao.in.net/

www.khumao.in.net

Hello its www



khumao.in.net

its just khumao