

Install with apt, yum

```
#for ubuntu
sudo apt-get update -y & sudo
apt-get install -y nginx & sudo
systemctl enable nginx & sudo
systemctl start nginx
#for Centos
sudo yum update -y & sudo yum
install -y epel-release & sudo
yum install -y nginx & sudo
systemctl enable nginx & sudo
systemctl start nginx
```

Creating a Virtual Hosts

```
# make copy of
/etc/nginx/nginx.conf and begin
edit current conf file
server{
listen 80;
server_name 192.168.1.1
root /sites /demo
}
```

file types/ mime.types

```
#in http section
include mime.types;
types{
text/css css;
text/html html;
}
```

Locations

```
#Preferential prefix match
location ^~ /hello {
return 200 'Hello';
}
#exact match
location = /hello {
return 200 'Hello';
}
```

Locations (cont)

```
> #regex match - case sensitive
location ~ /hello[1-9] {
return 200 'Hello';
}
#regex match - case insensitive
location ~* /hello[1-9] {
return 200 'Hello';
}
```

buffers and Timeouts

```
#Buffer size for POST
submissions
client_body_buffer_size
10K;
client_max_body_size 8m;
#Buffer size for HEADERS
client_header_buffer_size
1K;
#MAX time to receive client
header/body
client_body_timeout 12;
client_header_timeout 12;
#MAX time to keep a connection
open for
keepalive_timeout 15;
#MAX time for client
accept/receive a response
send_timeout 10;
#skip buffering for static files
sendfile 10;
#optimize sendfile packets
tcp_nopush 10;
```

Source code build

```
sudo apt-get update -y
wget http://nginx.org/download/nginx_source.tar.gz
tar -zxvf nginx_source.tar.gz
#for ubuntu#
```

Source code build (cont)

```
> sudo apt-get install -y build-essential
#install builder
sudo apt-get install -y libpcre3 libpcre3-dev
zlib1g zlib1g-dev libssl-dev
#for centos#
sudo yum install -y groupinstall "Develo-
pment Tools"
sudo yum install -y pcre pcre-devel zlib zlib-
devel openssl openssl-devel
# in unpacked location
./configure --help
sudo ./configure --sbin-path=/usr/bin/nginx -
--conf-path=/etc/nginx/nginx.conf --error-l-
og-path=/var/log/nginx/error.log --http-log-
path=/var/log/nginx/access.log --with-pcre --
pid-path=/var/run/nginx.pid --with-http_ssl-
_module
sudo make & sudo make install & nginx -V
```

Variables

```
set $weekend 'No';
if ($date_local ~ 'Satur -
day |Sunday'){
set $weekend 'Yes';}
$host, $http_host, $https,
$request_method, $request -
_uri, $scheme, $server_name,
$server_port, $args, $arg_name,
$uri
```

Rewrites & Redirects

```
# Simple redirect
location /logo{
return 307 /logo.png;
}
#simple rewrite
rewrite ^/user /(\w+) /(\s ome -
thing) /greet/$1 $2;
rewrite ^/rewrite me/ (.*)$ /$1
last;
```

try_files

```
root /var/www/main;  
try_files $uri $uri.html $uri/  
/fallback /in dex.html;
```

dynamic modules

```
load_module                                sudo apt-get update -y & sudo  
modules/nginx_http_image_filter_module;sudo get -y install php-fpm &  
http {server {listen = log o.p ngsudo enable php7.2-fpm  
image_filter rotate 180;}}              user www-data
```

headers&expires

```
location ~* \.(jpg|png|js) {
    access_log off;
    add_header Cache-Control
public;
    add_header Pragma public;
    add_header Vary Accept -En -
coding;
    expires 1M;}

```

Adding nginx service if source-code

```

Create file
/lib/systemd/system/nginx.service
add and edit for current passes:
[Unit]

Description=The NGINX HTTP and reverse proxy server
After=syslog.target network.target
Wants=remote-fs.target nss-lookup.target

[Service]

Type=forking
PIDFile=/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -n/nginx -t
ExecStart=/usr/sbin/nginx
ExecReload=/usr/sbin/nginx -s reload
ExecStop=/bin/kill -s QUIT $MAINPID

PrivateTmp=true

[Install]

```

Adding nginx service if source-code (cont)

> WantedBy=multi-user.target

<https://www.nginx.com/resources/wiki/start/topics/examples/systemd/>

PHP Processing

```

sudo apt-get update -y & sudo
sudo apt-get -y install php-fpm &
sudo enable php7.2-fpm
user www-data
server{
index index.php index.h tml;
    loc ation / {
        try _files $uri $uri/
=404;
    }
    loc ation ~\.php${
        #pass php request to the
php-fpm servic e(f astcgi)
        include fastcg i.conf
        fas tcg i_pass unix:/ -
run /ph p/p hp7.2- fpm.sock
    }
}
}

```

worker_processes

```
worker_processes auto;
events {
    worker_connections
1024;}
# you can test connection_p
er_core limit by 'ulimit -n'
command
pid /var/run/new nginx.pid;
```

Compressed Responses with gzip

```
http{
gzip on;
gzip_comp_level 4;
gzip_types /text/css /text/ -
  jav asc ript;}
```

FASTCGI Cache

```
#configure microcache (fastcgi)
fastcg i_c ach e_path /tmp/n -
gin x/cache levels=1:2 keys_z -
one =ZO NE_1:100m inacti -
ve=60m;

fastcg i_c ach e_key " $sc hem -
e$r equ est _me tho d$h ost -
$re que st_ uri ";

add_header X-Cache $upstr eam -
_ca che _st atus;

location ~\.php$ {

include fastcg i.conf;

fastcg i_pass unix:/ run /ph -
p/p hp7.2- fpm.sock;

#enabling cahe

fastcg i_cahe ZONE_1;

fastcg i_c ahe _valid 200 60m;

fastcg i_c ahe _valid 404 10m;

}
```

LoadBallancing

```
upstream balancer {
    server 192.16 8.1.1:80;
    server 192.16 8.1.2:80;}

server {
    listen 192.16 8.1.3:80;
    server_name your_site.com;

    error_log /var/log/nginx/your_site.com.error.log;

    location / {
        proxy_pass
http://balancer;
    }
}

ip_hash; # freeze connection
from one host to one server
least_conn; # connection to
server with less loading
```