Nginx basics Cheat Sheet

by dm0610 via cheatography.com/77761/cs/19035/

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-r elease & 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 192.16 8.1.1
root /sites /demo
}
```

file types/ mime.types

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

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 _bo dy_ buf fer _size
10K;
client _ma x_b ody _size 8m;
#Buffer size for HEADERS
client _he ade r_b uff er_size
1K;
#MAX time to receive client
header s/body
client _bo dy_ timeout 12;
client _he ade r_t imeout 12;
#MAX time to keep a connection
open for
keepal ive _ti meout 15;
#MAX time for client
accept /re ceive a response
send_t imeout 10;
#skip buffering for static files
sendfile 10;
#optimize sendfile packets
tcp_nopush 10;
```

Source code build

```
sudo apt-get update -y
wget http:/ /ng inx.or g/d own -
loa d/n gin x_c ur_ ver.tar.gz
tar -zxvf nginx_ cur _ve r.t -
ar.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 |Su nday'){
set $weekend 'Yes';}
$host, $http_ host, $https,
$reque st_ method, $reque -
st_uri, $scheme, $serve r_name,
$serve r_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 $2;
rewrite ^/rewr ite me/ (.*)$ /$1
last;
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

try_files

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

dynamic modules

```
load_module
modules/nginx_http_image_filter_module.so;
http{s erv er{ loc ati on= log o.p ng{
ima ge_ filter rotate 180;}}}
```

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]
Descri pti on=The NGINX HTTP and
reverse proxy server
After= sys log.target networ -
k.t arget remote -fs.target nss-
lo oku p.t arget
[Service]
Type=f orking
PIDFil e=/ run /ng inx.pid
ExecSt art Pre =/u sr/ sbi -
n/nginx -t
ExecSt art =/u sr/ sbi n/nginx
ExecRe loa d=/ usr /sb in/ nginx
-s reload
ExecSt op= /bi n/kill -s QUIT
$MAINPID
Privat eTm p=true
[Install]
```

Adding nginx service if source-code (cont)

> WantedBy=multi-user.target

https://www.nginx.com/resources/wiki/st-art/topics/examples/systemd/

PHP Processing

```
sudo apt-get update -y & 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 {
        wor ker _co nne ctions
1024;}
# yuo can test connec tio n_p -
er_core limit by 'ulimit -n'
command
pid /var/r un/ new _ng inx.pid;
```

Compressed Responses with gzip

```
http{
gzip on;
gzip_c omp _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 balanser {
      server 192.16 8.1.1:80;
      server 192.16 8.1.2:80;}
server {
      listen 192.16 8.1.3:80;
      ser ver _name your_s -
ite.com;
      err or_log /var/l og/ -
ngi nx/ you r_s ite.co m_e rro -
r.log;
      loc ation / {
            pro xy_pass
http:/ /ba lancer;
     }
}
ip_hash; # freeze connection
from one host to one server
least_ conn; # connection to
server with less-l oading
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com