

# NGINX / PHP / MARIADB INSTALL

This will just be a quick write up of the steps that can be used to setup a Nginx web server using the nginx repo, along with PHP from the Remi repo, and mariadb from the MariaDB repo. By using the Nginx and MariaDB repositories, I can have a bit of comfort knowing that the applications are clean, maintained, and patched. There are other methods out there to get PHP loaded in, but I find that the Remi repo does a pretty good job at maintaining the packages, and they also have a great collection of already built modules that just slip right in and work. So let me get started.

**ROCKY LINUX:** <https://rockylinux.org>

**NGINX:** <https://nginx.org>

**REMI PHP:** <https://rpms.remirepo.net>

**MARIADB:** <https://mariadb.com>

**EPEL:** <https://docs.stg.fedoraproject.org/en-US/epel/>

```
# ----- FIND A NICE PLACE TO WORK -----
```

```
cd /opt
```

```
# ----- SWAP CENTOS LOGOS, ADD EPEL AND PHP REPOS -----
```

```
dnf -y swap centos-logos-httpd rocky-logos-httpd
```

```
dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
dnf -y config-manager --set-enabled powertools
```

```
dnf -y install https://rpms.remirepo.net/enterprise/remi-release-8.rpm
```

```
dnf config-manager --enable remi
```

```
# ----- RESET DNF MODULES AND ENABLE REMI PHP-----
```

```
dnf -y module reset php
```

```
dnf -y module reset nginx
```

```
dnf -y module reset httpd
```

```
dnf -y module disable php*
```

```
dnf -y module disable composer*
```

```
dnf -y module enable php:remi-8.2
```

```
dnf -y module enable composer
```

```
# ----- ADD MARIADB REPO -----
```

```
wget https://downloads.mariadb.com/MariaDB/mariadb_repo_setup
```

```
chmod +x mariadb_repo_setup
```

```
./mariadb_repo_setup --mariadb-server-version="mariadb-11.4"
```

```
rm -f mariadb_repo_setup
```

```
# ----- ADD NGINX REPO FILE -----
```

```
cat > /etc/yum.repos.d/nginx.repo << "EOF"
```

```
[nginx-stable]
```

```
name=nginx stable repo
```

```
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
```

```
gpgcheck=1
```

```
enabled=0
```

```
gpgkey=https://nginx.org/keys/nginx_signing.key
```

```
module_hotfixes=true
```

```
[nginx-mainline]
```

```
name=nginx mainline repo
```

```
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
```

```
gpgcheck=1
```

```
enabled=1
```

```
gpgkey=https://nginx.org/keys/nginx_signing.key
```

```
module_hotfixes=true
```

```
EOF
```

```
# ----- ENABLE MAINLINE REPO -----
```

```
dnf config-manager --enable nginx-mainline
```

```
# ----- UPDATE THE SYSTEM AND INSTALL DEVELOPMENT TOOLS -----
```

```
dnf -y update --refresh
```

```
dnf -y group install "Development Tools"
```

```
# ----- INSTALL NGINX, MARIADB, AND A COUPLE PHP MODULES -----
```

```
dnf -y install \  
nginx \  
php-mysqlnd php-pecl-mysql \  
MariaDB-server MariaDB-client mariadb-tools
```

```
# ----- ENABLE AND START MARIADB AND RUN THE SECURE SCRIPT -----
```

```
systemctl enable mariadb  
systemctl start mariadb
```

```
source mariadb-secure-installation
```

```
### I ANSWER THE QUESTIONS INITIALLY LIKE SO...
```

```
Enter current password for root (enter for none): 'enter'
```

```
Switch to unix_socket authentication [Y/n] 'n'
```

```
Change the root password? [Y/n] 'n'
```

```
Remove anonymous users? [Y/n] 'y'
```

```
Disallow root login remotely? [Y/n] 'n'
```

```
Remove test database and access to it? [Y/n] 'y'
```

```
Reload privilege tables now? [Y/n] 'y'
```

```
### RUN THE FOLLOWING TO ENABLE REMOTE ROOT ACCESS
```

```
### WARNING: CHANGE THE PASSOWRD FROM 'PASSWORD'
```

```
#
```

```
# mariadb
```

```
# CREATE USER 'root'@'%' IDENTIFIED BY 'PASSWORD';
```

```
# GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' WITH GRANT OPTION;
```

```
# FLUSH PRIVILEGES;
```

```
# exit
```

```
#
```

```
# ----- FIX UP SOME PERMISSIONS AND ENABLE SERVICES-----
```

```
chown -R nginx:nginx /usr/share/nginx/html  
chown -R nginx:nginx /var/lib/php/session/  
chown -R root:nginx /var/lib/php/{opcache,wsdlcache}  
chown -R nginx:nginx /var/log/{nginx,php-fpm}
```

```
systemctl enable nginx
systemctl enable php-fpm
```

```
# ----- QUICK CONFIG FOR PHP-FPM -----
```

```
mv /etc/php-fpm.d/www.conf /etc/php-fpm.d/www.bak
```

```
cat > /etc/php-fpm.d/www.conf << "EOF"
```

```
[www]
```

```
user = nginx
```

```
group = nginx
```

```
listen.owner = nginx
```

```
listen.group = nginx
```

```
listen.mode = 0660
```

```
listen = /var/run/php-fpm/www.sock
```

```
listen.allowed_clients = 127.0.0.1
```

```
pm = dynamic
```

```
pm.max_children = 25
```

```
pm.start_servers = 6
```

```
pm.min_spare_servers = 6
```

```
pm.max_spare_servers = 18
```

```
pm.process_idle_timeout = 10s
```

```
pm.max_requests = 250
```

```
pm.status_path = /status
```

```
slowlog = /var/log/php-fpm/www-slow.log
```

```
php_admin_value[error_log] = /var/log/php-fpm/www-error.log
```

```
php_admin_flag[log_errors] = on
```

```
php_value[session.save_handler] = files
```

```
php_value[session.save_path] = /var/lib/php/session
```

```
php_value[soap.wsdl_cache_dir] = /var/lib/php/wsdlcache
```

```
; Default Value: clean env
```

```
clear_env = no
```

```
;env[HOSTNAME] = $HOSTNAME
```

```
;env[PATH] = /usr/local/bin:/usr/bin:/bin
```

```
;env[TMP] = /tmp
;env[TMPDIR] = /tmp
;env[TEMP] = /tmp
```

EOF

```
# ----- QUICK DEFAULT SITE CONFIG WITH PHP ENABLED -----
mv /etc/nginx/conf.d/default.conf /etc/nginx/conf.d/default.bak
cp /etc/nginx/nginx.conf /etc/nginx/nginx.bak
cat > /etc/nginx/nginx.conf << "EOF"
# CUSTOM CONF
```

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
```

```
events {
    worker_connections 1024;
}
```

```
http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';
```

```
    access_log    /var/log/nginx/access.log;
```

```
    sendfile      on;
    tcp_nopush     on;
    tcp_nodelay    on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
```

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

```
    include       /etc/nginx/conf.d/*.conf;
```

```
    server {
```

```

listen    80 default_server;
server_name _;

root      /usr/share/nginx/html;

include   /etc/nginx/default.d/*.conf;

location  / {

}

error_page 404 /404.html;
    location = /40x.html {
}

error_page 500 502 503 504 /50x.html;
    location = /50x.html {
}
}
EOF

# ----- DROP IN A PHPINFO FILE -----
cat > /usr/share/nginx/html/phpinfo.php << "EOF"
<?php
    phpinfo();
?>
EOF

# ----- START NGINX AND PHP-FPM -----

systemctl start nginx
systemctl start php-fpm

```

You should be able to hit the default nginx welcome page and phpinfo.php to validate php is working.

http://<ip / hostname>/  
http://<ip / hostname>/phpinfo.php

Hope this was helpful! :)

