

Zoneminder FreeBSD Installation and configuration

Starting with github instructions: <https://raw.githubusercontent.com/freebsd/freebsd-ports/master/multimedia/zoneminder/files/README.FreeBSD>

New installs

=====

ZoneMinder requires a MySQL (or MySQL forks) database backend and a http server, capable to execute PHP and CGI scripts.

To simplify things, we assume, that you use MySQL and NGINX on the same server.

Above there is an assumption that you have FEMP installed. On my first attempt I've installed and configured FEMP, followed by pkg zoneminder to install the application, but noticed that zm installation removes some applications from the FEMP stack, so inverted the process installing zoneminder first and FEMP later.

1. Zoneminder port installation

Command:

```
root@freebsd-1:~ # pkg install zoneminder
```

Output:

Updating FreeBSD repository catalogue...

FreeBSD repository is up-to-date.

All repositories are up-to-date.

The following 33 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:

```
zoneminder: 1.25.0_7
pcre: 8.38_1
php56-session: 5.6.22
php56: 5.6.22
libxml2: 2.9.3
mysql56-client: 5.6.30 ← Where's mysql server?
libedit: 3.1.20150325_2
gcc: 4.8.5_2
mpc: 1.0.3
gmp: 5.1.3_3
mpfr: 3.1.3_1
binutils: 2.25.1_1,1
gcc-ecj: 4.5
jpeg-turbo: 1.4.2
ffmpeg0: 0.7.17_2,1
libtheora: 1.1.1_6
libvorbis: 1.3.5,3
libogg: 1.3.2_1,4
libvpx: 1.5.0_1
xvid: 1.3.4,1
schroedinger: 1.0.11_4
orc: 0.4.25
freetype2: 2.6.3
libx264: 0.144.2533_1
opencv-core: 2.4.9_3
php56-json: 5.6.22
sudo: 1.8.16_1
```

*netpbm: 10.35.98
jbigkit: 2.1_1
jasper: 1.900.1_16
tiff: 4.0.6_1
png: 1.6.21
php56-mysql: 5.6.22*

*The process will require 528 MiB more space.
93 MiB to be downloaded.*

Proceed with this action? [y/N]: y

2. FEMP

2.1 – Installation with pkg

Command:

```
root@freebsd-1:~ # pkg install nginx mysql56-server php56 php56-mysql
```

Output:

*Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 2 package(s) will be affected (of 0 checked):*

New packages to be INSTALLED:

*nginx: 1.8.1_3,2
mysql56-server: 5.6.30*

*The process will require 85 MiB more space.
8 MiB to be downloaded.*

Proceed with this action? [y/N]:

2.2 – Enable Services

Verify what's installed

Command:

```
root@freebsd-1:~ # grep rcvar /usr/local/etc/rc.d/*
```

Output:

*/usr/local/etc/rc.d/ffserver0:rcvar=ffserver0_enable
/usr/local/etc/rc.d/mysql-server:rcvar=mysql_enable
/usr/local/etc/rc.d/nginx:rcvar=nginx_enable
/usr/local/etc/rc.d/php-fpm:rcvar=php_fpm_enable
/usr/local/etc/rc.d/webmin:rcvar=webmin_enable
/usr/local/etc/rc.d/zoneminder:rcvar=zoneminder_enable*

2.3.– Enable FEMP for configuration

a) Edit rc.conf (use vi, nano, joe, or whatever you like)

Command:

```
root@freebsd-1:~ # joe /etc/rc.conf
```

Add below lines to the end and save file.

```
mysql_enable="YES"
nginx_enable="YES"
php_fpm_enable="YES"
```

Will look like the one below:

```
hostname="freebsd-1"
ifconfig_em0="DHCP"
sshd_enable="YES"
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable
dumpdev="AUTO"
webmin_enable=YES
mysql_enable="YES"
nginx_enable="YES"
php_fpm_enable="YES"
```

3. Configure PHP

3.1. Edit php-fpm.conf

Command:

```
root@freebsd-1:~ # joe /usr/local/etc/php-fpm.conf
```

- Find the line: **listen = 127.0.0.1:9000**
- Replace with: **listen = /var/run/php-fpm.sock**

Will look like this:

```
; The address on which to accept FastCGI requests.
; Valid syntaxes are:
; 'ip.add.re.ss:port' - to listen on a TCP socket to a specific IPv4 addr
;                      a specific port;
; '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6 addr
;                      a specific port;
; 'port' - to listen on a TCP socket to all IPv4 addresses o
;                      specific port;
; '[:]:port' - to listen on a TCP socket to all addresses
;                      (IPv6 and IPv4-mapped) on a specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /var/run/php-fpm.sock
```

- Two sessions below you will find the permission for unix socket, **uncomment the three first commands** to look like this:

```
; Set permissions for unix socket, if one is used. In Linux, read/write
; permissions must be set in order to allow connections from a web server. Man
; BSD-derived systems allow connections regardless of permissions.
; Default Values: user and group are set as the running user
; mode is set to 0660
listen.owner = www
listen.group = www
listen.mode = 0660
```

- **Save and close the file**

3.2 – Create php.ini

- **Copy the sample production file**

Commands:

```
root@freebsd-1:/ # cd /usr/local/etc
root@freebsd-1:/usr/local/etc # cp php.ini-production php.ini
```

- Edit the file and set `cgi.fix_pathinfo=0`

Command:

```
root@freebsd-1:/ # joe php.ini
```

Will look like this:

```
; cgi.fix_pathinfo provides *real* PATH_INFO/PATH_TRANSLATED support for CGI.
; previous behaviour was to set PATH_TRANSLATED to SCRIPT_FILENAME, and to not
; what PATH_INFO is. For more information on PATH_INFO, see the cgi specs. S
; this to 1 will cause PHP CGI to fix its paths to conform to the spec. A set
; of zero causes PHP to behave as before. Default is 1. You should fix your
; to use SCRIPT_FILENAME rather than PATH_TRANSLATED.
; http://php.net/cgi.fix-pathinfo
cgi.fix_pathinfo=0
```

- **Set your date time zone**
 - Uncomment `date.timezone` and type your time zone within single quotes (').

Note: Check your time zone here: <http://php.net/manual/en/timezones.php>

I'm in EST, so will set to 'America/New_York' and will look like this:

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = 'America/New_York'
```

- **Save the file**

3.3 - Start php

Command:

```
root@freebsd-1:/usr/local/etc # service php-fpm start
```

Output:

Performing sanity check on php-fpm configuration:

[16-Jun-2016 10:23:48] NOTICE: configuration file /usr/local/etc/php-fpm.conf test is successful

Starting php_fpm.

4. Configure MySQL

4.1. - Start MySQL

Command:

```
root@freebsd-1:/usr/local/etc # service mysql-server start
```

Output:

Starting mysql.

4.2. Configure MySQL

Command:

```
root@freebsd-1:/usr/local/etc # mysql_secure_installation
```

Output:

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current password for the root user. If you've just installed MySQL, and you haven't set the root password yet, the password will be blank, so you should just press enter here.

Enter current password for root (enter for none):

- **Press ENTER and set your new password**

Output: (Instructions in **RED** below)

*Change the root password? [Y/n] **Y***

*New password: **<type your password>***

*Re-enter new password: **<retype your password for confirmation>***

Password updated successfully!

Reloading privilege tables..

... Success!

By default, a MySQL installation has an anonymous user, allowing anyone to log into MySQL without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

- **Press ENTER on all following prompts** (Instructions in **RED** below)

*Remove anonymous users? [Y/n] **<ENTER>***

... Success!

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] **<ENTER>**

... Success!

By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

Remove test database and access to it? [Y/n] **<ENTER>**

- Dropping test database...

... Success!

- Removing privileges on test database...

... Success!

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

Reload privilege tables now? [Y/n] **<ENTER>**

... Success!

All done! If you've completed all of the above steps, your MySQL installation should now be secure.

Thanks for using MySQL!

Cleaning up...

4.3. - Restart MySQL

Command:

```
root@freebsd-1:/usr/local/etc # service mysql-server restart
```

Output:

Stopping mysql.

Waiting for PIDS: 1427.

Starting mysql.

5. Configuring nginx

5.1. Start nginx

Command:

```
root@freebsd-1:/usr/local/etc # service nginx start
```

Output:

Performing sanity check on nginx configuration:

nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok

nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful

Starting nginx.

5.2. - Create nginx.conf

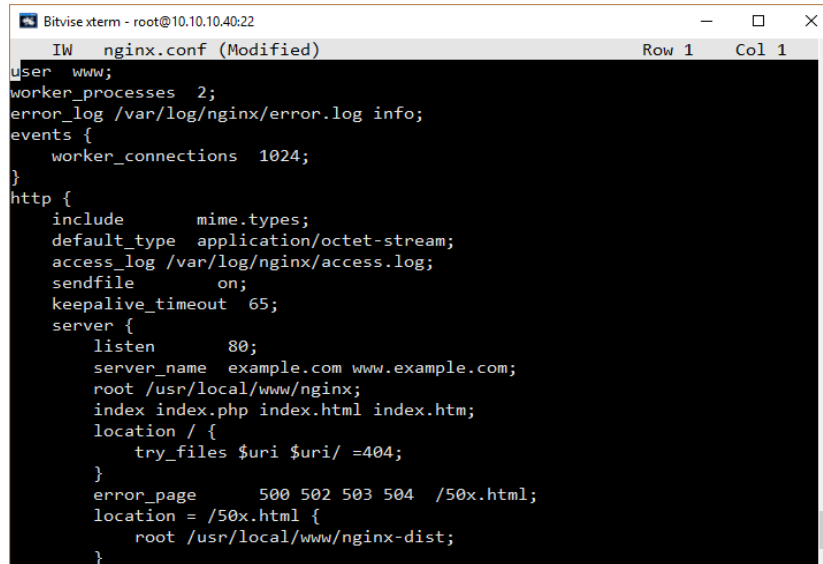
Copy sample below to the new file been created:

```
user www;
worker_processes 2;
error_log /var/log/nginx/error.log info;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    access_log /var/log/nginx/access.log;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name example.com www.example.com;
        root /usr/local/www/nginx;
        index index.php index.html index.htm;
        location / {
            try_files $uri $uri/ =404;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root /usr/local/www/nginx-dist;
        }
        location ~ \.php$ {
            try_files $uri =404;
            fastcgi_split_path_info ^(.+\.php)(/.+)$;
            fastcgi_pass unix:/var/run/php-fpm.sock;
            fastcgi_index index.php;
            fastcgi_param SCRIPT_FILENAME $request_filename;
            include fastcgi_params;
        }
    }
}
```

Commands:

```
root@freebsd-1:/usr/local/etc # cd /usr/local/etc/nginx
root@freebsd-1:/usr/local/etc/nginx # mv nginx.conf nginx.conf-original
root@freebsd-1:/usr/local/etc/nginx # joe nginx.conf
```

- You will have something like:



```
Bitvise xterm - root@10.10.10.40:22
IW nginx.conf (Modified) Row 1 Col 1
user www;
worker_processes 2;
error_log /var/log/nginx/error.log info;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    access_log /var/log/nginx/access.log;
    sendfile on;
    keepalive_timeout 65;
    server {
        listen 80;
        server_name example.com www.example.com;
        root /usr/local/www/nginx;
        index index.php index.html index.htm;
        location / {
            try_files $uri $uri/ =404;
        }
        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
            root /usr/local/www/nginx-dist;
        }
    }
}
```

- Save the file

5.3. - Create logs directory and empty files

Commands:

```
root@freebsd-1:/usr/local/etc/nginx # mkdir -p /var/log/nginx
root@freebsd-1:/usr/local/etc/nginx # touch /var/log/nginx/access.log
root@freebsd-1:/usr/local/etc/nginx # touch /var/log/nginx/error.log
```

5.4. - Configuring document root

```
root@freebsd-1:/usr/local/etc/nginx # rm /usr/local/www/nginx
root@freebsd-1:/usr/local/etc/nginx # mkdir /usr/local/www/nginx
root@freebsd-1:/usr/local/etc/nginx # cp /usr/local/www/nginx-dist/index.html /usr/local/www/nginx
```

5.5. - Create a simple php file for test

- Create a file named info.php
- Paste the following line on it and save the file

```
<?php phpinfo(); ?>
```

Command:

```
root@freebsd-1:/usr/local/etc/nginx # joe /usr/local/www/nginx/info.php
```


5.6. - Test nginx configuration

Command:

```
root@freebsd-1:/usr/local/etc/nginx # nginx -t
```

Output:

```
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
```

Note: If you follow all steps above you shouldn't have any errors here. If you have any errors, redo your steps as all above instructions and outputs were captured from a real configuration on a FreeBSD installed from: FreeBSD-10.3-RELEASE-amd64-dvd1.

5.7. Restart nginx

Command:

```
root@freebsd-1:/usr/local/etc/nginx # service nginx restart
```

Output:

```
Performing sanity check on nginx configuration:
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
Stopping nginx.
Waiting for PIDS: 1659.
Performing sanity check on nginx configuration:
nginx: the configuration file /usr/local/etc/nginx/nginx.conf syntax is ok
nginx: configuration file /usr/local/etc/nginx/nginx.conf test is successful
Starting nginx.
```

6. - Testing nginx with a browser

6.1. - Getting the local IP for the server

There are 1,000 ways. The one I like as find easy if with ifconfig

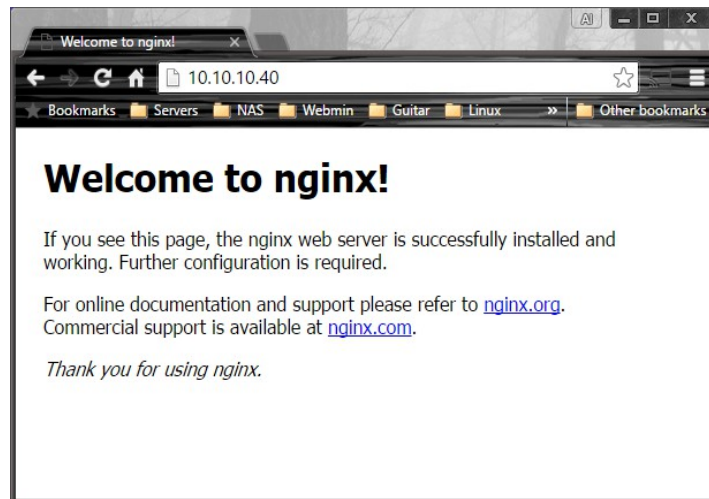
Command:

```
root@freebsd-1:/usr/local/etc/nginx # ifconfig
```

Output:

```
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
ether 08:00:27:8e:96:13
inet 10.10.10.40 netmask 0xfffff00 broadcast 10.10.10.255
nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
media: Ethernet autoselect (1000baseT <full-duplex>)
status: active
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
options=600003<RXCSUM,TXCSUM,RXCSUM_IPV6,TXCSUM_IPV6>
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x2
inet 127.0.0.1 netmask 0xff000000
nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
```

6.2. - Open a browser and type the IP address above and you should get the nginx page:



6.3. - Testing php

Open a browser and type your server ip address, followed by **/info.php**. (i.e. 10.10.10.40/info.php in my case). You will get the page below:



7. Back to github readme

We've configured FEMP and MySQL, nginx, and php are running as below:

```
PID USERNAME  THR PRI NICE  SIZE  RES STATE  TIME  WCPU COMMAND
729 root       1  20  0 86580K 5128K select 0:01 0.00% sshd
1652 mysql    21  52  0 638M   451M sigwai 0:01 0.00% mysqld
602 root       1  20  0 77388K 13288K select 0:00 0.00% perl
642 root       1  20  0 24152K 3392K select 0:00 0.00% sendmail
1237 root      1  20  0 39844K 6660K kqread 0:00 0.00% php-fpm
1906 www       1  20  0 28820K 5520K kqread 0:00 0.00% nginx
735 root       1  20  0 23600K 2668K pause   0:00 0.00% csh
442 root       1  20  0 14520K 1412K select 0:00 0.00% syslogd
1990 root      1  20  0 21948K 2448K RUN     0:00 0.00% top
649 root       1  20  0 16624K 1516K nanslp  0:00 0.00% cron
731 root       1  29  0 23600K 2164K pause   0:00 0.00% csh
1546 mysql     1  52  0 17096K 2076K wait    0:00 0.00% sh
287 root       1  40  0 14656K 1436K select 0:00 0.00% dhclient
1238 www       1  20  0 39844K 7160K accept 0:00 0.00% php-fpm
704 root       1  52  0 14520K 1452K ttyin   0:00 0.00% getty
362 root       1  20  0 13628K 564K select 0:00 0.00% devd
1907 www       1  20  0 28820K 5576K kqread 0:00 0.00% nginx
733 root      1  28  0 50384K 4048K select 0:00 0.00% sftp-serve
```

Step 1 from README

1. Preliminary steps

1.1 Install databases/mysql56-server

You may choose your favourite method - ports or packages here.

FreeBSD default setting use STRICT_TRANS_TABLES sql_mode. It's mandatory to disable it. Edit your my.cnf or create new in /var/db/mysql

The following example works with ZoneMinder quite well

```
[server]
skip-networking
skip-name-resolve
innodb_flush_method = O_DIRECT
skip-innodb_doublewrite
innodb_file_per_table
```

Copy and paste session highlighted in yellow to the new file: /var/db/mysql/my.cnf

Command:

```
root@freebsd-1: / # joe /var/db/mysql/my.cnf
```

Now, enable and start MySQL

```
sysrc mysql_server_enable="YES"
service mysql-server start
```

We already have mysql running, so restart it with the command below:

Command:

```
root@freebsd-1: / # service mysql-server restart
```

Output:

Stopping mysql.

Waiting for PIDS: 1652.

Starting mysql.

1.2 Install www/nginx

We provide an example for an HTTP install, however, you should use HTTPS if you plan to expose your installation to the public. There are plenty guides how to do it and security/letsencrypt.sh is a good way to get a valid SSL certificate.

Nginx is already installed and running, so only need to change the server session, so let's save the working version and create a new one with the configuration below:

Commands:

```
root@freebsd-1:~ # cd /usr/local/etc/nginx
```

```
root@freebsd-1:~ # mv nginx.conf nginx.conf-working
```

Create a new config with the configuration below:

```
user www;
worker_processes 2;
error_log /var/log/nginx/error.log info;
events {
    worker_connections 1024;
}
http {
    include mime.types;
    default_type application/octet-stream;
    access_log /var/log/nginx/access.log;
    sendfile on;
    keepalive_timeout 65;
    server {
        root /usr/local/www/zoneminder;
        try_files $uri $uri/ /index.php$is_args$args;
        index index.php;

        location = /cgi-bin/nph-zms {
            include fastcgi_params;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            fastcgi_pass unix:/var/run/cgiwrap/cgiwrap.sock;
        }

        location ~ /\.php$ {
            include fastcgi_params;
            fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
            fastcgi_pass unix:/var/run/php-fpm.sock;
        }

        location /api {
            rewrite ^/api/(.+)$ /api/index.php?p=$1 last;
        }
    }
}
```

Command:

```
root@freebsd-1:/usr/local/etc/nginx # joe nginx.conf
```

1.3 Install www/fcgiwrap

As NGINX lacks it's own CGI wrapper, we need external one. Please note that ZoneMinder's montage page use simultaneous access to all cameras, so you need to use at least as many fcgiwrap workers as your number of cameras. The following example assumes you have 4.

```
Enable and start FcgiWrap
    sysrc fcgiwrap_enable="YES"
    sysrc fcgiwrap_user="www"
    sysrc fcgiwrap_flags="-c 4"
```

That I've not installed, so:

Commands:

```
root@freebsd-1:/usr/local/etc/nginx # pkg install fcgiwrap
```

Output:

```
Updating FreeBSD repository catalogue...
FreeBSD repository is up-to-date.
All repositories are up-to-date.
The following 2 package(s) will be affected (of 0 checked):
```

New packages to be INSTALLED:

```
fcgiwrap: 1.1.0_3
fcgi-devkit: 2.4.0_5
```

181 KiB to be downloaded.

*Proceed with this action? [y/N]: **Y ← Press Y and ENTER***

Output:

```
Fetching fcgiwrap-1.1.0_3.txz: 100% 11 KiB 11.5kB/s 00:01
Fetching fcgi-devkit-2.4.0_5.txz: 100% 170 KiB 174.1kB/s 00:01
Checking integrity... done (0 conflicting)
[1/2] Installing fcgi-devkit-2.4.0_5...
[1/2] Extracting fcgi-devkit-2.4.0_5: 100%
[2/2] Installing fcgiwrap-1.1.0_3...
[2/2] Extracting fcgiwrap-1.1.0_3: 100%
```

Note: I don't see fcgiwraper beed started in the README.FREEBSD. Should I? I guess so, based on the lines below:

Enable and start FcgiWrap

```
    sysrc fcgiwrap_enable="YES"
    sysrc fcgiwrap_user="www"
    sysrc fcgiwrap_flags="-c 4"
```

I'll be using 10 cameras, so the last command in my case will be: sysrc fcgiwrap_flags="-c 10"

Let try it:

Commands:

```
root@freebsd-1:/usr/local/etc/nginx # cd /
root@freebsd-1:/ # sysrc fcgiwrap_enable="YES"
fcgiwrap_enable: -> YES
root@freebsd-1:/ # sysrc fcgiwrap_user="www"
fcgiwrap_user: -> www
root@freebsd-1:/ # sysrc fcgiwrap_flags="-c 10"
fcgiwrap_flags: -> -c 10
root@freebsd-1:/ # service fcgiwrap start
Starting fcgiwrap.
```

1.4 PHP is installed as a dependency to ZoneMinder. However, you should tweak some of it's settings.
Edit /usr/local/etc/php-fpm.conf and set

```
listen = /var/run/php-fpm.sock
listen.owner = www
listen.group = www
env[PATH] = /usr/local/bin:/usr/bin:/bin
```

We did install php and tested with our FEMP initial installation, but have not set the environment path, so edit the file and uncomment that line:

Command:

```
root@freebsd-1:/usr/local/etc/nginx # joe /usr/local/etc/php-fpm.conf
```

Will look like this:

```
; Pass environment variables like LD_LIBRARY_PATH. All $VARIABLEs are taken fr
; the current environment.
; Default Value: clean env
;env[HOSTNAME] = $HOSTNAME
env[PATH] = /usr/local/bin:/usr/bin:/bin
```

1.5 ZoneMinder constantly keeps the last N frames from its cameras to preserve them when alarm occurs. This can be a performance hog if placed on spindle drive. The best practice is put it on tmpfs. See <https://www.freebsd.org/cgi/man.cgi?query=tmpfs> for more information.

ZoneMinder will use /tmp for default. If you plan to change it, see ZM_PATH_MAP setting.

Mapping /tmp to tmpfs is actually a recommended step under FreeBSD. Edit /etc/fstab and add the following:

```
tmpfs          /tmp          tmpfs
rw,nosuid,mode=01777    0            0
```

So let's change fstab

Edit /etc/fstab and add the recommended line from the READ

Command:

```
root@freebsd-1:/ # joe /etc/fstab
```

Mine looks like this:

```
# Device    Mountpoint  FStype Options Dump  Pass#
/dev/ada0p2 /           ufs    rw    1    1
/dev/ada0p3 none        swap   sw    0    0
tmpfs      /tmp        tmpfs  rw,nosuid,mode=01777 0    0
```

2. ZoneMinder installation

Connect to MySQL under root and create zm user and populate database.

Now the fun begin!

- Create user and database (*Note the for this example I'm not changing the user or password. You should change it to something only your know, but there are other steps involved to change other configuration files I won't be covering here*)

Commands and outputs:

```
root@freebsd-1:/ # mysql -u root -p
```

```
Enter password: ← type your sql password
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 1
```

```
Server version: 5.6.30 Source distribution
```

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.

```
mysql> CREATE DATABASE zm;
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> GRANT ALL PRIVILEGES ON zm.* TO 'zmuser'@'localhost' IDENTIFIED BY 'zmpass';
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> FLUSH PRIVILEGES;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> quit;
```

```
Bye
```

```
root@freebsd-1:/ #
```

- Populate the database

From the README.FREEBSD file the command is:

```
root@freebsd-1:/ # mysql -u root -p zm < /usr/local/share/zoneminder/db/zm_create.sql  
/usr/local/share/zoneminder/db/zm_create.sql: No such file or directory.
```

Maybe I made a mistake on my installation ... looking for zm_create.sql file

Command:

```
root@freebsd-1:/ # whereis zm_create.sql  
zm_create.sql:
```

Ok ... Let's try another way ...

```
root@freebsd-1:/ # find / -name zm_create.sql  
/usr/local/share/doc/zoneminder/zm_create.sql
```

It says it is @ /usr/local/share/doc/zoneminder

Adjusting the command to:

```
root@freebsd-1:/ # mysql -u root -p zm < /usr/local/share/doc/zoneminder/zm_create.sql  
Enter password:  
root@freebsd-1:/ #
```

It seems to be worked ...

2.1 If you have chosen to change the ZoneMinder MySQL credentials to something other than zmuser/zmpass then you must now edit /usr/local/etc/zm.conf.

Change ZM_DB_USER and ZM_DB_PASS to the values you created in the previous step.

Additionally, you must also edit /usr/local/www/zoneminder/api/app/Config/database.php in a similar manner. Scroll down and change login and password to the values you created in the previous step.

This is what I was referring above when created the database, user, etc ...

```
Enable and start ZoneMinder
    sysrc zoneminder_enable="YES"
    service zoneminder start
```

Almost there ...

Commands and outputs:

```
root@freebsd-1:/ # sysrc zoneminder_enable="YES"
zoneminder_enable: -> YES
```

```
root@freebsd-1:/ # service zoneminder start
/usr/local/etc/rc.d/zoneminder: ERROR: zoneminder: mysqltest command failed
root@freebsd-1:/ #
```

Didn't work... Maybe someone can find my error and help writing up this how-to.