

FAMP

Welcome to FreeBSD!

1. Boot FreeBSD [default]
2. Boot FreeBSD with ACPI disabled
3. Boot FreeBSD in Safe Mode
4. Boot FreeBSD in single user mode
5. Boot FreeBSD with verbose logging
6. Escape to loader prompt
7. Reboot

Select option, [Enter] for default
or [Space] to pause timer 8



Nagios on FreeBSD

Apache2.4, PHP 8.0, Nagios 4 on FreeBSD 12.3, 13.0 and 13.1 (FAMP)

June 10, 2022 femtopc.com, updated October 18, 2022

Synopsis: Nagios Core 4 provides monitoring of all mission-critical infrastructure components including applications, services, operating systems, network protocols, systems metrics, and network infrastructure.

https://www.femtopc.com/shell_scripts/FAMP/Nagios/

Requirements:

1. FreeBSD server, on discrete hardware or running in a virtual machine.
2. Script designed for server with clean FreeBSD install, no other software installed.
3. Disk requirement: After installation, <3GB are used, including the vanilla FreeBSD installation.
4. Must be run with 'root' access, or as a user that 'su -' to root.
5. Server must have Internet access.
6. Script file must run in /root directory.
7. Depending on Internet speed and server processing power, installation is complete in under 10 minutes.

Step-by-step:

1. Login to FreeBSD server as root, or 'su -' to root.

2. Fetch the FAMP package

```
fetch https://www.femtopc.com/shell_scripts/FAMP/Nagios/install.gz
```

3. Unzip the download

```
Tar -xvf install.gz
```

4. Run the shell program and answer when prompted.

```
./nagios.sh
```

5. Script will check to make sure you are running as root, in the /root directory and have Internet access. It will then check for any Apache, MySQL and PHP packages and exit if found. The install version of PHP will be 8.0. After packages are installed, the configuration will be automatically done. The php.ini can choose from the 'production' source before the file is modified. View the logfile.log to troubleshoot any errors.

Here is a breakdown of the comments and commands that are run in the script.

```
# install Apache2.4
pkg install -y apache24

# install PHP80
pkg install -y php80

# install PHP80-extensions
pkg install -y php80-extensions

# install Apache mod_php80
pkg install -y mod_php80

# These are what the configuration routines do
# modify the /usr/local/etc/apache24/httpd.conf file
# make a backup and then edit
cd /usr/local/etc/apache24
cp httpd.conf httpd.conf.original
ee httpd.conf

# set the ServerAdmin, in the script it is set to root. You may also set a webmaster email here.
# set the ServerName to the IP address of server. A domain name could be set here if available.
# set the second time AllowOverride None, where None is capital "N" to All. This will allow .htaccess files
to work.
# set DirectoryIndex from just index.html to index.php index.htm index.html to capture PHP files
# setup Apache to run PHP by creating an Includes/php.conf file
cd Includes
echo "<FilesMatch \"\\.php$\">" > php.conf
echo "SetHandler application/x-httpd-php" >> php.conf
echo "</FilesMatch>" >> php.conf
echo "<FilesMatch \"\\.phps$\">" >> php.conf
echo "SetHandler application/x-httpd-php-source" >> php.conf
echo "</FilesMatch>" >> php.conf
# return to our root directory
cd /root
```

```
# Make Apache start automatically when FreeBSD boots
sysrc apache24_enable="yes"

# Startup the Apache Server
service apache24 start

# Now configure PHP
cd /usr/local/etc

# choose production (will not display web page errors)
cp php.ini-production php.ini

# edit the php.ini file
ee php.ini

# find where memory_limit = 128M and increase to 1024M
# find where post_max_size = 8M and set to 32M (or set your limit here)
# find where upload_max_filesize = 2M and set to 1G (or your limit here)
# find where max_file_uploads = 20 and increase to 80
# lastly, find ;date.timezone and uncomment (delete ;) and enter timezone (IMPORTANT!!)
# hint: find your server's setting by doing "cat /var/db/zoneinfo"
# after changes to php.ini, Apache must be restarted
service apache24 restart

# at this point you have a working FAP server
cd /root

# install nagios 4.4.6
pkg install -y nagios4
sysrc nagios_enable="YES"
cd /usr/local/etc/nagios
mv cgi.cfg-sample cgi.cfg
mv nagios.cfg-sample nagios.cfg
mv resource.cfg-sample resource.cfg
cd objects
mv commands.cfg-sample commands.cfg
mv contacts.cfg-sample contacts.cfg
```

```
mv localhost.cfg-sample localhost.cfg
mv printer.cfg-sample printer.cfg
mv switch.cfg-sample switch.cfg
mv templates.cfg-sample templates.cfg
mv timeperiods.cfg-sample timeperiods.cfg
mv windows.cfg-sample windows.cfg
cd /usr/local/etc/apache24
sed -i " 's/#LoadModule cgid_module/LoadModule cgid_module/g' /usr/local/etc/apache24/httpd.conf
sed -i " 's/#LoadModule cgi_module/LoadModule cgi_module/g' /usr/local/etc/apache24/httpd.conf
sed -i " -e 's|<IfModule alias_module>|<IfModule alias_module>\n \n #added for nagios server\n\n
<Directory \"/usr/local/www/nagios/^\n Require all granted\n AllowOverride all\n php_flag engine on\n
php_admin_value open_basedir /usr/local/www/nagios/:/var/spool/nagios/\n </Directory>\n ScriptAlias
/nagios/cgi-bin \"/usr/local/www/nagios/cgi-bin/\n Alias /nagios \"/usr/local/www/nagios/\n <Directory
"/usr/local/www/nagios/cgi-bin/\n Options ExecCGI\n </Directory>\n|1'
/usr/local/etc/apache24/httpd.conf
sed -i " 's/#AddHandler cgi-script/AddHandler cgi-script/g' /usr/local/etc/apache24/httpd.conf
cd /usr/local/www/nagios
# create nagios admin user and password for web login
touch /usr/local/www/nagios/.htpasswd >> /root/logfile.log
echo "AuthName \"Nagios Login\"
AuthType Basic
AuthUserFile /usr/local/www/nagios/.htpasswd
require user nagiosadmin" > /usr/local/www/nagios/.htaccess
htpasswd -c /usr/local/www/nagios/.htpasswd nagiosadmin ←- enter a password
chmod 644 .htpasswd
chmod 644 .htaccess
cd /root
#Restart Services
service apache24 restart
service nagios start
# This completes the Nagios server install and setup. Login at:
http://your_server_ip_address/nagios
username: nagiosadmin with your chosen password
<end>
```