

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



Apache2.4, MySQL 8.0, PHP 8.1 on FreeBSD 12.3, 12.4, 13.0 and 13.1 (FAMP)

December 11, 2022 femtopc.com

Synopsis: Nextcloud Hub is the industry-leading, fully open-source, on-premises content collaboration platform. Teams access, share and edit their documents, chat and participate in video calls and manage their mail and calendar and projects across mobile, desktop and web interfaces. This document is a supplement to FemtoPC's "FAMP" installation script file for FreeBSD 12.3, 12.4, 13.0 and 13.1. It is customized to install MySQL 8.0 and PHP 8.1 and will grab the latest Nextcloud community server.

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

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, only 3.1GB 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/Nextcloud/install.gz
```

3. Unzip the download

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

4. Run the shell program and answer when prompted.

```
./nc.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 versions of MySQL and PHP are fixed. After packages are installed, the configuration will be automatically done. The php.ini can choose either the 'production' source or the 'developer' source before the file is modified. A password will be requested for MySQL root user. Please make it secure. 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 MySQL80-Server
pkg install -y mysql80-server
# install PHP81
pkg install -y php81
# install PHP81-extensions
pkg install -y php81-extensions
# Extra extensions for Nextcloud will be installed
pkg install -y php81-bz2
pkg install -y php81-curl
pkg install -y php81-exif
pkg install -y php81-fileinfo
pkg install -y php81-gd
pkg install -y php81-intl
pkg install -y php81-ldap
pkg install -y php81-mbstring
pkg install -y php81-pdo_mysql
pkg install -y php81-zip
pkg install -y php81-zlib
pkg install -y php81-pecl-imagick
pkg install -y php81-pecl-apcu
pkg install -y php81-bcmath
pkg install -y php81-gmp
# install Apache mod_php81
pkg install -y mod_php81
```

```
# These are what the configuration routines done
# 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 /usr/local/etc/apache24/Includes
ee php.conf

<FilesMatch "\.php$" >
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$" >
    SetHandler application/x-httpd-php-source
</FilesMatch># 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

# Setup to run MySQL server
# Make Server start automatically when FreeBSD boots.
sysrc mysql_enable="yes"

# start the mysql server for the first time
service mysql-server start

# MySQL80 has no password set for root
# Set your new password; it MUST BE in single quotes
/usr/local/bin/mysqladmin -u root password 'your new password'

# For MySQL 80, just press Enter when prompted for a password, as none was assigned
```

```
# Now configure PHP
cd /usr/local/etc

# choose a development or a production environment
# development (displays errors in web pages)
cp php.ini-development php.ini

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

# edit the php.ini file
ee php.ini

# find where output_buffering = 4096 and instead set to off, required for Nextcloud
# find ;user_ini.cache_ttl = 300 line and insert after, apc_enable_cli = 1, for NC
# 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 FAMP server
cd /root

# This is the specific installation and configuration for Nextcloud
# Get the latest Nextcloud files from Nextcloud.com
fetch "https://download.nextcloud.com/server/releases/latest.tar.bz2"

# Install the bzip2 utility
pkg install -y bzip2

# Extract the downloaded .bz2 file to a tar
bunzip2 -fk latest.tar.bz2

# Extract the tar file to nextcloud folder
tar -xpf nextcloud.tar
```

```
# Make it owned by www
chown -R www:www nextcloud

# Get our Apache webroot ready by creating a 'nextcloud' directory
cd /usr/local/www/apache24/data
mkdir nextcloud
chown -R www:www nextcloud

# Create and own the core data directory
cd /usr/local/www/apache24
mkdir core
chown -R www:www core
chmod 760 core

# return to /root and extracted download directory
cd /root/nextcloud

# Copy all files to the ~webroot/nextcloud directory
cp -R * /usr/local/www/apache24/data/nextcloud

# Install nextcloud manually (substitute the mysql root user password for $mysqlrootpw)
su -m www -c 'php occ maintenance:install --database "mysql" --database-name "nextcloud" --database-user "root" --database-pass "$mysqlrootpw" --admin-user "admin" --admin-pass "admin" --data-dir "/usr/local/www/apache24/core"'

# setup to use cron as background task
su -m www -c 'php occ background:cron'

# setup the config file
cd /usr/local/www/apache24/data/nextcloud/config
ee config.php

# insert a line after '0 => 'localhost'', as '1 => '$ip'', where $ip is the server's IP address
# change 'http://localhost', to 'http://$ip/nextcloud', where $ip is the server's ip
# add two more lines after: be sure to keep the single quotes and commas
# 'check_for_working_wellknown_setup', => false
# 'memcache.local' => '\OC\Memcache\APCu',
```

#example of a config.php file with previous changes:

```
<?php
$CONFIG = array (
    'passwordsalt' => 'mi4g6nVhaNI57qjewpLks5WvPyDWxo',
    'secret' => '5tL9fEHT9fhRiwOl4hyZ2MQ+mxZyuhzTRvYScgSfNSpIHMW7',
    'trusted_domains' =>
    array (
        0 => 'localhost',
        1 => '192.168.0.73',
    ),
    'datadirectory' => '/usr/local/www/apache24/core',
    'dbtype' => 'mysql',
    'version' => '25.0.2.3',
    'overwrite.cli.url' => 'http://192.168.0.73/nextcloud',
    'check_for_working_wellknown_setup' => false,
    'memcache.local' => '\OC\Memcache\APCu',
    'dbname' => 'nextcloud',
    'dbhost' => 'localhost',
    'dbport' => '',
    'dbtableprefix' => 'oc_',
    'mysql.utf8mb4' => true,
    'dbuser' => 'oc_admin',
    'dbpassword' => ']qRJOCw6x8e1Cix.-;4S6sy.C.GOE.',
    'installed' => true,
    'instanceid' => 'oc4irl5l0cwg',
);
```

setup the crontab, we'll do this without using the vi editor, which crontab would call up

create a temporary file **ctmp**

```
echo "SHELL=/bin/sh" >> ctmp
echo "PATH=/usr/local/bin" >> ctmp
echo "*/5 * * * * php /usr/local/www/apache24/data/nextcloud/cron.php" >> ctmp
```

use the crontab command as web user 'www', pointing to file contents, **then delete ctmp**

```
crontab -u www ctmp
```

```
rm -f ctmp
```

restart cron

```
service cron restart
```

Restart Apache one last time

```
service apache2 restart
```

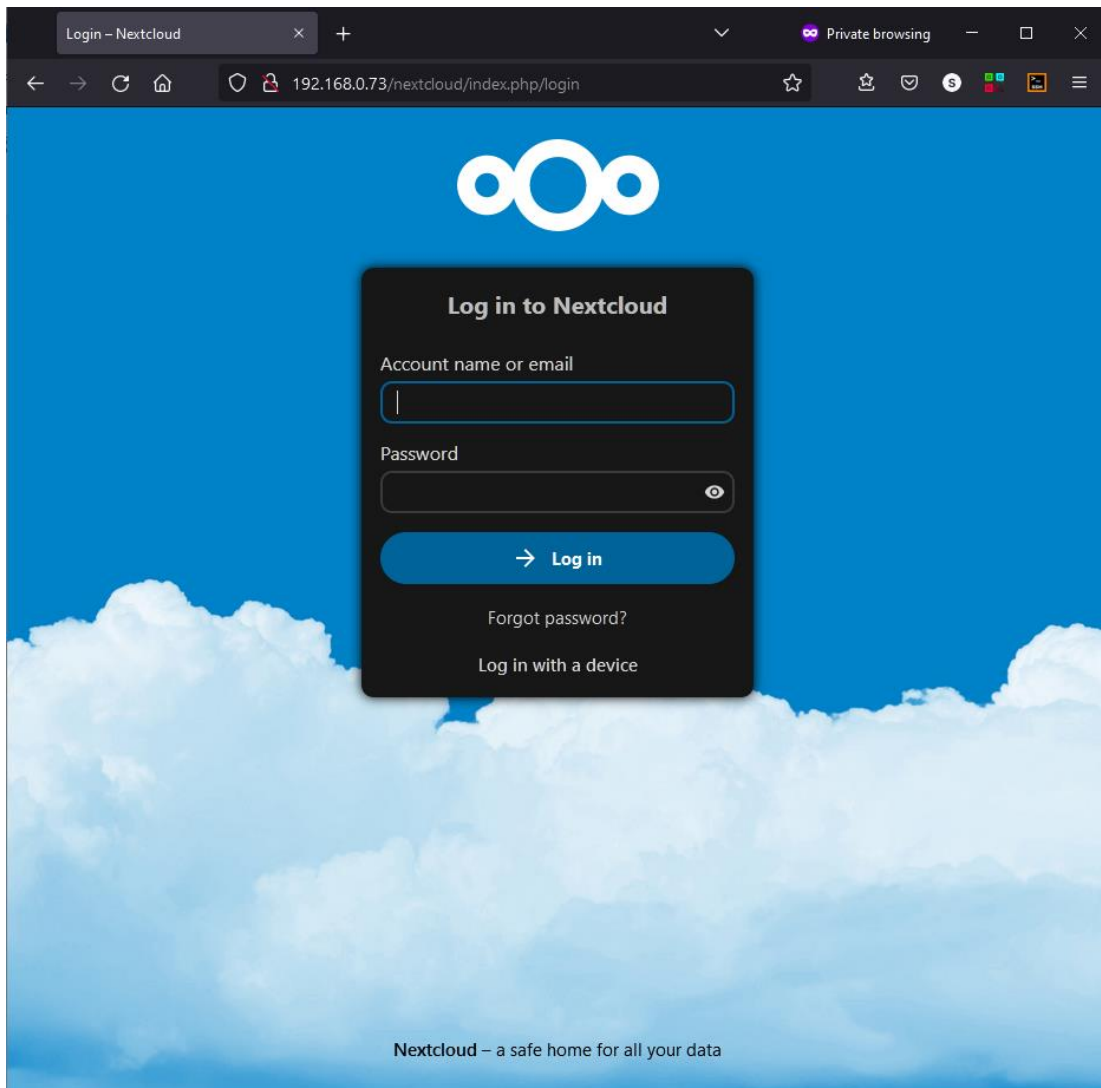
run Nextcloud's background job manually one time

```
cd /usr/local/www/apache24/data/nextcloud
```

```
su -m www -c 'php cron.php'
```

```
cd /root
```

- # Open a browser and connect to the web server's IP address
- # A working server will reply with "It works!"
- # Login Nextcloud by opening <http://{ip of server}/nextcloud/index.php>
- # Remember the following and input is required:
- # Name: admin
- # Password: admin
- # Change the admin password through the web interface



<end>