

PHP

1. [How to check the PHP modules installed?](#)
2. [PHP configuration file](#)
3. [How to modify the allowed size for uploaded files?](#)
4. [How to modify the execution time?](#)
5. [How to install eAccelerator module?](#)
6. [How to install APC module?](#)
 - 6.1.1. [How to check the APC cache usage?](#)
7. [How to install OAuth module?](#)
8. [How to install Memcache module?](#)
9. [How to install Memcached module using libmemcached library?](#)
10. [How to install Tidy module?](#)
11. [How to install Mongo module?](#)
12. [How to install mod_geoip2 module?](#)
13. [How to install OCI8 module?](#)
14. [How to install PDO_IBM module?](#)
 - 14.1. [Windows](#)
 - 14.2. [Linux](#)
15. [How to install FreeTDS and MSSQL module?](#)
16. [How to send email via external SMTP account](#)
17. [How to send email using sendmail?](#)
18. [How to change the date or timezone](#)
19. [How in install IonCube extension?](#)
20. [How in install Zend Guard Loader extension?](#)
21. [How to install Solr module?](#)
22. [How to install New Relic for PHP?](#)
23. [How to install OPcache for PHP?](#)
24. [How to enable OpenSSL extension?](#)
25. [How to install SSH2 for PHP?](#)
26. [How to enable Imagick extension?](#)
27. [How to enable XCache extension?](#)
 - 27.1. [How to install XCache admin?](#)
28. [How to configure NetBeans with Bitnami LAMP, MAMP or WAMP stacks?](#)
29. [How to modify the PHP memory settings?](#)

How to check the PHP modules installed?

The easiest way to test your PHP installation is to create a test script using the `phpinfo()` function. Open your favorite text editor and type:

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

Save the file as "phptest.php" in `/install_dir/apache2/htdocs/`. Make sure Apache is up and running, open a browser and type <http://localhost/phptest.php>. You should then see a screen showing detailed information about the PHP version you are using.

You can also running the following command:

```
$ /install_dir/php/bin/php -m
```

PHP configuration file

The PHP configuration file allows you to configure the modules enabled, the email settings or the size of the upload files. It is located at:

BitNami Cloud Hosting and Virtual Machines

Ubuntu: `/opt/bitnami/php/etc/php.ini`

Amazon Linux / Red Hat Enterprise: `/etc/php.ini`

BitNami Installers

linux_platform.png On Linux: `/install_dir/php/etc/php.ini`

win_platform.png On Windows: `/install_dir/php/php.ini`

mac_platform.png On OS X: `/install_dir/php/etc/php.ini`

How to modify the allowed size for uploaded files?

You can modify the following option in the [php.ini](#) file to increase the allowed size for uploads:

```
; Maximum size of POST data that PHP will accept.  
post_max_size = 16M  
...
```

```
; Maximum allowed size for uploaded files.  
upload_max_filesize = 16M
```

If you have enabled PHP-FPM (enabled by default in **Cloud Images** and **VMs**) you need to restart PHP-FPM running the following command:

```
sudo /opt/bitnami/ctlscript.sh restart php-fpm
```

Note: For native installers replace /opt/bitnami with your current installation directory.

Otherwise, you need to restart the Apache server:

```
sudo YOUR_INSTALLATION_DIRECTORY/ctlscript.sh restart apache
```

How to modify the execution time?

This parameter sets the timeout for Apache to run a script. You can modify the following option in the [php.ini](#) file.

```
max_execution_time = 120      ; Maximum execution time of each script, in
```

How to install eAccelerator module?

[eAccelerator](#) is a free open-source PHP accelerator & optimizer. It increases the performance of PHP scripts by caching them in their compiled state, so that the overhead of compiling is almost completely eliminated. It also optimizes scripts to speed up their execution.

These are the steps to install this module on top of BitNami Stacks in Ubuntu Linux. For Windows systems you can download the module at http://www.sitebuddy.com/PHP/Accelerators/eAccelerator_windows_binaries_builds

- Install the following packages:

```
$ sudo apt-get update  
$ sudo apt-get install build-essential libtool autoconf unzip
```

- Download the latest source code from the [eAccelerator web page](#), copy to the server, uncompress it and compile the module

```
$ unzip eaccelerator-*.zip  
$ export PHP_AUTOCONF=/usr/bin/autoconf
```

```
$ export PHP_PREFIX=/opt/bitnami/php
$ cd eaccelerator-0.9.6.1
$ /opt/bitnami/php/bin/phpize
$ ./configure --enable-eaccelerator=shared --with-php-config=$PHP_PREFIX/bin/php-config
$ make
$ sudo make install
```

- Then, enable the module in the [php.ini](#) file:

```
...
extension=eaccelerator.so
...
```

- And check it, you can see something similar to this

```
$ php -version
```

```
Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies
    with eAccelerator v0.9.5-beta2, Copyright (c) 2004-2006 eAccelerator
```

- You can configure this module in the [php.ini](#) file. You can find more info at <http://eaccelerator.net/wiki/InstallFromSource>

How to install APC module?

[APC](#) is an Alternative PHP Cache module. Most of the BitNami Stacks already include this module and you can check it at `"/install_dir/php/lib/php/extensions"` folder. If the `apc.so` module exists, you only should enable it in the [php.ini](#) file.

In case this module does not exist. You can install it manually. These are the steps to install it in an Ubuntu machine.

- Install the following packages:

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget
```

- Download the latest source code from the web page, uncompress it and compile the module.

```
$ wget http://pecl.php.net/get/APC
$ tar -zxf APC*
$ export PHP_AUTOCONF=/usr/bin/autoconf
$ export PHP_PREFIX=/opt/bitnami/php
```

```
$ cd APC-*
$ /opt/bitnami/php/bin/phpize
$ ./configure --enable-apc --enable-apc-mmap --with-php-config=$PHP_PREF
$ make
$ sudo make install
```

- Then, enable the module uncommenting the following lines in the [php.ini](#) file (remove ';' before each of them):

```
...
extension = apc.so
apc.enabled = 1
...
```

How to check the APC cache usage?

APC ships a PHP application to check the memory usage. It is only necessary to copy the "apc.php" file from the APC source files to the Apache htdocs folder.

```
$ wget http://pecl.php.net/get/APC
$ tar -zxvf APC*
$ cp APC*/apc.php /installldir/apache2/htdocs
```

Then you can check the APC cached files at http://your_domain/apc.php

How to install OAuth module?

[OAuth](#) is an authorization protocol built on top of HTTP which allows applications to securely access data without having to store usernames and passwords.

These are the steps to install it in an Ubuntu machine.

- Install the following packages:

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget
```

- Download the latest source code from the web page, uncompress it and compile the module.

```
$ wget http://pecl.php.net/get/oauth-1.2.2.tgz
$ tar -zxf oauth*
$ export PHP_AUTOCONF=/usr/bin/autoconf
$ export PHP_PREFIX=/opt/bitnami/php
$ cd oauth-*
$ /opt/bitnami/php/bin/phpize
$ ./configure
$ make
$ sudo make install
```

- Then, enable the module adding the following line in the [php.ini](#) file:

```
...
extension = '/opt/bitnami/php/lib/php/extensions/oauth.so'
...
```

- Finally, you can check if the extension is loaded with the following command:

```
$ php -i | grep -i oauth
```

How to install Memcache module?

Memcache module provides handy procedural and object oriented interface to memcached, highly effective caching daemon, which was especially designed to decrease database load in dynamic web applications.

If this module is not in your Stack, you can install it manually. These are the steps to install it in an Ubuntu machine.

- Install the following packages:

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget
```

- Download the latest source code from the web page, uncompress it and compile the module. I have downloaded the 2.2.4 version that it is the latest stable version at the time this writing.

```
$ wget http://pecl.php.net/get/memcache-2.2.4.tgz
$ tar -zxf memcache-2.2.4.tgz
$ export PHP_AUTOCONF=/usr/bin/autoconf
```

```
$ export PHP_PREFIX=/opt/bitnami/php
$ cd memcache-2.2.4
$ /opt/bitnami/php/bin/phpize
$ ./configure --enable-memcache --with-zlib-dir=/opt/bitnami/common
$ make
$ sudo make install
```

- Then, enable the module in the [php.ini](#) file:

```
...
extension=memcache.so
...
```

How to install Memcached module using libmemcached library?

Memcached is a high-performance, distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic web applications by alleviating database load. This extension uses the libmemcached library to provide an API for communicating with memcached servers.

If this module is not in your Stack, you can install it manually. These are the steps to install it in an Ubuntu Virtual Machine (if you have installed it using a Native Installer, substitute the paths where /opt/bitnami appears with your installation directory).

- Install the following packages:

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget libcl
```

- Download the latest source code from the web page, uncompress it and compile the module. I have downloaded the libmemcached 1.0.16 version and the 2.1.0 module version, that are the latest stable and working version at the time this was written.

After tried to use libmemcached 1.0.17, we received an error message during the compilation:

```
libtest/server_container.cc:241: error:
'HAVE_MEMCACHED_BINARY' was not declared in this scope
```

This seems to be a bug in that libmemcached version (<https://bugs.launchpad.net/libmemcached/+bug/1164440>), so we decided to use libmemcached 1.0.16

```
$ wget https://launchpad.net/libmemcached/1.0/1.0.16/+download/libmemcached-1.0.16.tar.gz
$ tar -zxf libmemcached-1.0.16.tar.gz
$ cd libmemcached-1.0.16
$ ./configure --prefix=/opt/bitnami/common
$ make
$ sudo make install
$ cd ~/
$ wget http://pecl.php.net/get/memcached-2.1.0.tgz
$ tar -zxf memcached-2.1.0.tgz
$ export PHP_AUTOCONF=/usr/bin/autoconf
$ export PHP_PREFIX=/opt/bitnami/php
$ cd memcached-2.1.0
$ /opt/bitnami/php/bin/phpize
$ ./configure --enable-memcached --with-zlib-dir=/opt/bitnami/common --with-libmemcached-dir=/opt/bitnami/common
$ make
$ sudo make install
```

- Then, enable the module in the [php.ini](#) file:

```
...
extension=memcached.so
...
```

- If you are on Bitnami Cloud Hosting, you can enable memcache server running this commands:

```
mv /opt/bitnami/memcached/scripts/ctl.sh.disabled /opt/bitnami/memcached/scripts/ctl.sh
/opt/bitnami/ctlscript.sh start memcached
```

- If you got an error, check if the Stack bundle memcached server:

```
/opt/bitnami/ctlscript.sh status memcached
```

- Usually, if you don't see "memcached" status, it is because you are using a stack that doesn't bundle memcached server or you are not using Bitnami Cloud Hosting. To install and start it, you must run:

```
apt-get install memcached
/etc/init.d/memcached start
```

- Restart the Stack services:

```
/opt/bitnami/ctlscript.sh restart
```

- Finally, check if memcached is running properly. To do it, create a .php file on your htdocs folder with this code and access to it using the browser:

```
<?php
    $mc = new Memcached();
    $mc->addServer("127.0.0.1", 11211);

    $result = $mc->get("test_key");

    if($result) {
        echo $result;
    } else {
        echo "No data on Cache. Please refresh page pressing F5 ";
        $mc->set("test_key", "test data pulled from Cache!") or die ("Failed to set");
    }
?>
```

How to install Tidy module?

Tidy is a binding for the Tidy HTML clean and repair utility which allows you to not only clean and otherwise manipulate HTML documents, but also traverse the document tree.

If this module is not in your Stack, you can install it manually. These are the steps to install it in an Ubuntu Virtual Machine or AMI.

- Install the following packages (they are already in the machine in recent versions)

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget
```

- Download the latest source code from the web page, uncompress it and compile the module.

```
$ wget http://tidy.cvs.sourceforge.net/viewvc/tidy/tidy/?view=tar
```

- Uncompress it and build the libraries

```
$ cd /home/bitnami
$ tar -xzf tidy-tidy.tar.gz
$ cd tidy/build/gmake
$ make
```

- Now download the PHP source code version that you are currently using from <http://www.php.net> and uncompress it. Then build the shared module for tidy:

```
$ tar -xzf php-<version>.tar.gz
$ cd php/ext/tidy
$ ./phpize
$ ./configure --with-tidy=/home/bitnami/tidy
$ make
$ sudo make install
```

How to install Mongo module?

Mongo PHP module is a MongoDB driver. If this module is not in your Stack, you can install it manually. These are the steps to install it in an Ubuntu machine.

```
$ sudo pecl install mongo
...
Build process completed successfully
Installing '/opt/bitnami/php/lib/php/extensions/<wbr>mongo.so'
install ok: channel://pecl.php.net/mongo-<wbr>1.4.5</wbr><wbr>
configuration option "php_ini" is not set to php.ini location
You should add "extension=mongo.so" to php.ini</wbr></wbr>
```

Then, enable the module in the [php.ini](#) file and restart Apache:

```
...
extension=mongo.so
...
```

How to install mod_geoip2 module?

[Mod_geoip2](#) module embeds GeoIP database lookups into the Apache web server. It is only capable of looking up the IP of a client that

connects to the web server, as opposed to looking up arbitrary addresses.

If this module is not in your Stack, you can install it manually. These are the steps to install it in an Ubuntu Virtual Machine or AMI.

- Install the following packages (they are already installed in the machine in recent versions)

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget
```

- In this case it is necessary the development files for libgeoip:

```
$ sudo apt-get install libgeoip-dev
```

- Download the latest source code from the web page, uncompress it and compile the module.

```
$ wget http://www.maxmind.com/download/geoip/api/mod_geoip2/mod_geoip2-l
```

- Uncompress it and build the libraries

```
$ tar -xzf mod_geoip2-latest.tar.gz
$ cd mod_geoip2_*
$ sudo apxs -i -a -lGeoIP -c mod_geoip.c
```

- The module is now available in /opt/bitnami/apache2/modules/mod_geoip.so. It is also automatically included in your Apache configuration. However, you will need to enable GeoIP in the Apache configuration adding the line "GeoIPEnable On" at /opt/bitnami/apache2/conf/httpd.conf. You can find other specific settings at http://dev.maxmind.com/geoip/mod_geo...onfiguration-2
- Restart the Apache server.

```
$ sudo /opt/bitnami/ctlscript.sh restart apache
```

How to install OCI8 module?

Since LAMP/MAMP/WAMP version 5.4.34, 5.5.18 and 5.6.2, Bitnami stacks ship the 'oci8' module already compiled for each platform. This module requires to download and install InstantClient libraries from Oracle.

For enabling the 'oci8' module in a LAMP/MAMP/WAMP stack it is necessary to enable it in the 'php.ini' file. In this section we will use '/opt/bitnami/' as the installation directory, you should specify your installation directory.

1. Enable 'oci8' module in the '/opt/bitnami/php/etc/php.ini':
2. If you try to check if the module is working you can get the following error:

```
$ /opt/bitnami/php/bin/php -m
```

PHP Warning: PHP Startup: Unable to load dynamic library '/php/lib/php/'

3. Download InstantClient Basic package version 11.2.0.4 for your platform from Oracle page <http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>
4. Unzip the package in your home folder, f.e. '/home/bitnami/instantclient_11_2'

5. Add the following environment variable at the end of your '/opt/bitnami/scripts/setenv.sh' file:

For Linux:

```
LD_LIBRARY_PATH=/home/bitnami/instantclient_11_2:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

For OS X:

```
DYLD_LIBRARY_PATH=/Users/bitnami/instantclient_11_2:$LD_LIBRARY_PATH
export DYLD_LIBRARY_PATH
```

6. Restart your servers and you can check it is working properly if the following command does not throw any error:

```
$ /opt/bitnami/php/bin/php -m
```

How to install PDO_IBM module?

From LAMP/MAMP/WAMP version 5.4.34, 5.5.18 and 5.6.2, Bitnami stacks include the 'PDO_IBM' module already compiled for Linux and Windows platforms.

For enabling this module in a LAMP/MAMP/WAMP stack it is necessary to enable it in the 'php.ini' file. In this section we will use '/opt/bitnami/' as the installation directory, you should specify your installation directory.

Windows

- Enable 'PDO_IBM' module in the '/opt/bitnami/php/etc/php.ini', adding the following line:

```
extension=php_pdo_IBM.dll
```

- If you run "YOUR_INSTALLDIR/php/php.exe -m" command you will get an error pointing that DB2CLI.dll is missing. To fix this issue you will need to install the Data Server client for DB2 from IBM. You can download the installer for the client from the following link: <http://www-01.ibm.com/support/docview.wss?uid=swg27016878>
- Restart your servers through the graphical manager and then check if your extension is loaded properly running the following command:

```
YOUR_INSTALLDIR/php/php.exe -m | findstr IBM
```

Linux

- Enable 'PDO_IBM' module in the '/opt/bitnami/php/etc/php.ini', adding the following line:

```
extension=pdo_IBM.so
```

- You can try to check if the module is already enabled with the following command:

```
$ /opt/bitnami/php/bin/php -m
```

PHP Warning: PHP Startup: Unable to load dynamic library '/opt/bitnami/

- Download Data Server Driver Package for your platform from IBM page https://www14.software.ibm.com/webapp/iwm/web/reg/pick.do?source=swg-idsdpds&lang=en_US
- Unzip the package in your home folder, f.e. '/home/bitnami/' and install it running the following commands:

```
cd /home/bitnami/dsdriver
bash installDSDriver
```

- Add the following environment variable at the end of your '/opt/bitnami/scripts/setenv.sh' file:

```
LD_LIBRARY_PATH=/home/bitnami/dsdriver/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

- Restart your servers and you can check it is working properly if the following command does not throw any error:

```
$ /opt/bitnami/php/bin/php -m | grep ibm
```

How to install FreeTDS and MSSQL module?

FreeTDS is a set of libraries for Unix and Linux that allows your programs to natively talk to Microsoft SQL Server and Sybase databases. It is necessary to install the MSSQL module for PHP.

Since LAMP version 5.4.25, this module is already included in LAMP so it is not necessary to follow the steps below, just only enable the module in the php.ini file.

These are the steps to install FreeTDS and MSSQL module in a BitNami Ubuntu Virtual Machine or AMI.

- Load the build environment (and add -liconv to the LDFLAGS)

```
$ . /opt/bitnami/scripts/build-setenv.sh
$ export LDFLAGS="$LDFLAGS -liconv"
```

- Install FreeTDS.

```
$ wget ftp://ftp.astron.com/pub/freetds/stable/freetds-stable.tgz
$ tar xzf freetds-stable.tgz
$ cd freetds-0.91/
$ ./configure --enable-msdblib --prefix=/usr/local/freetds
$ make
$ sudo make install
$ cd ..
```

- Install MSSQL module for PHP.

Compile the mssql extension and copy it to the PHP directory. Note that you should download the same PHP version that you are using. You can check the current PHP version with the "php -v" command.

```
$ wget http://us.php.net/distributions/php-5.4.23.tar.bz2
$ tar jxf php-5.4.23.tar.bz2
$ cd php-5.4.23/ext/mssql/
$ phpize
$ ./configure --with-mssql=/usr/local/freetds
$ make
$ sudo make install
```

The last instruction will copy the mssql.so library into /opt/bitnami/php/lib/php/extensions/

Now you can edit /opt/bitnami/php/etc/php.ini and add the line extension=mssql.so

How to send email via external SMTP account

If you want to make sure your emails are delivered properly it is better to configure your PHP script or PHP based application with the external SMTP account. In the following example we will show you how to do it for Gmail account. It is enough to set proper host and username variables. Please note that the variables in your script/application may be different. You need Mail and Net_Smtp2 libraries installed to execute the script below:

```
$ sudo /opt/bitnami/php/bin/pear install pear/Net_Smtp2 pear/Mail-1.2.0
```

Note that if these packages are already installed in your system you see the messages below when executing that command:

```
Ignoring installed package pear/Net_Smtp2
Ignoring installed package pear/Mail
Nothing to install
```

Configuring the external SMTP account:

```
<?php
require_once "Mail.php";
```

```

$from = "myaccount@gmail.com";
$to = 'test@mytest.com';

$host = "ssl://smtp.gmail.com:465";
$username = 'myaccount@gmail.com';
$password = 'mypassword';

$subject = "test";
$body = "test";

$headers = array ('From' => $from, 'To' => $to, 'Subject' => $subject);
$smtp = Mail::factory('smtp',
    array ('host' => $host,
        'auth' => true,
        'username' => $username,
        'password' => $password));

$mail = $smtp->send($to, $headers, $body);

if (PEAR::isError($mail)) {
    echo($mail->getMessage());
} else {
    echo("Message successfully sent!\n");
}

?>

```

How to send email using sendmail?

If you want to use sendmail instead of an external SMTP server, you can configure it if you uncomment this line:

```
sendmail_path = "env -i /usr/sbin/sendmail -t -i"
```

in the PHP configuration file: */opt/bitnami/php/etc/php.ini*

How to change the date or timezone

The default timezone is "America/Los_Angeles" but you can change it in the [php.ini](#):

```
[Date]
```

```
date.timezone = "America/Los_Angeles"
```

You can see all available timezones at <http://php.net/manual/en/timezones.php>

How in install IonCube extension?

You can download the pre-compiled version and copy the .so file to php extensions directory. If you are on Linux you can use the "wget" command. In this example we are going to download the Linux 64 bit version.

```
$ wget http://downloads2.ioncube.com/loader_downloads/ioncube_loaders_lin_x86-64.tar.gz
$ tar xvzf ioncube_loaders_lin_x86-64.tar.gz
$ sudo cp ioncube/ioncube_loader_lin_5.4.so /opt/bitnami/php/lib/php/extensions/
```

Add the following line to php.ini:

```
zend_extension = /opt/bitnami/php/lib/php/extensions/ioncube_loader_lin_5.4.so
```

Then restart PHP-fpm and the Apache server.

If you are using a non-Ubuntu server (CentOS or Amazon Linux) PHP version could be different. You can check the PHP version that you have in the machine with the following command "php -v"

How in install Zend Guard Loader extension?

Zend Guard is a free runtime application that enables PHP to run the scripts encoded by Zend Guard. It can be freely used by anyone looking to run encoded applications. You can download the binaries extensions from the Zend Guard [site](#). In this example we are going to install the 64 bit version.

Once you download it and copy to the machine run the following commands:

```
$ tar -xzvf ZendGuardLoader-*-PHP-5.4-linux-glibc23-x86_64.tar.gz

$ sudo cp ZendGuard*/php-5.4.x/ZendGuardLoader.so /opt/bitnami/php/lib/php/extensions/
```

And add the following entry at the end of the php.ini file:

```
zend_extension = /opt/bitnami/php/lib/php/extensions/ZendGuardLoader.so
```

You can check the module was correctly installed with the following command:

```
$ php -m | grep Zend
```

How to install Solr module?

The Apache Solr PHP extension is an extremely fast, light-weight, feature-rich library that allows PHP applications to communicate easily and efficiently with Apache Solr server instances using an object-oriented API.

If this module is not in your Stack, you can install it manually. These are the steps to install it in an Ubuntu machine.

- Install the following packages:

```
$ sudo apt-get update
$ sudo apt-get install build-essential libtool autoconf unzip wget
```

- Download the latest source code [from the web page](#), uncompress it and compile the module:.

```
$ wget http://pecl.php.net/get/solr-1.0.2.tgz
$ tar -zxf solr-1.0.2.tgz
$ export PHP_AUTOCONF=/usr/bin/autoconf
$ export PHP_PREFIX=/opt/bitnami/php
$ cd solr-1.0.2
$ /opt/bitnami/php/bin/phpize
$ ./configure --enable-solr --with-curl=/opt/bitnami/common --with-libxml
$ make
$ sudo make install
```

- Then, enable the module in the [php.ini](#) file:

```
...
extension=solr.so
...
```

How to install New Relic for PHP?

New Relic is a Web application in real-time that run in cloud, on-premise, or hybrid environments. These are the steps to configure New Relic for PHP with Bitnami.

- Download the tarball from https://download.newrelic.com/php_agent/release/2
- Uncompress it in your machine:

```
$ tar -xzf newrelic-php5*.tar.gz
```

- It is necessary to apply the following patch before running the installer:

```
$ cd newrelic*
$ sed -i 's/${pi_bin} -n/${pi_bin}/' newrelic-install
```

- Run the New Relic install script:

```
$ sudo ./newrelic-install
```

- Check that the module was correctly installed at "/opt/bitnami/php/lib/php/extensions/newrelic.so"
- Add the newrelic.so module in the php configuration file. You can add the following at the end of the file:

```
$ sudo nano /opt/bitnami/php/etc/php.ini
```

```
extension=newrelic.so
```

- Check that it was properly added to php. You can run "php -m" and see that newrelic extension is enabled
- Restart Apache server

```
$ sudo /opt/bitnami/ctlscript.sh restart apache
```

- Check the New relic service is running in the machine:

```
$ sudo /etc/init.d/newrelic-daemon status
New Relic Daemon: newrelic-daemon is running
```

How to install OPCache for PHP?

You can use OPCache instead APC. To do it, please, follow this steps:

- Disable APC:

```
sudo sed -i 's/extension=apc.so;/extension=apc.so/g' /opt/bitnami/php/etc
```

- Install ZendOpcache:

```
sudo pecl install channel://pecl.php.net/ZendOpcache-7.0.3
```

- Change the opcache.so permissions:

```
sudo chmod ugo+x /opt/bitnami/php/lib/php/extensions/opcache.so
```

Finally, you only need to enable opcache. To do it you can run the sed command, or do it manually

- Sed command:

```
sudo sed -i 's/;extension=apc.so;/extension=apc.so\nzend_extension=/opt
```

- If you want to do it manually, you will need to edit */opt/bitnami/php/etc/php.ini*. After *extension=apc.so*, add this lines:

```
zend_extension=/opt/bitnami/php/lib/php/extensions/opcache.so
opcache.memory_consumption=128
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=4000
opcache.revalidate_freq=60
opcache.fast_shutdown=1
opcache.enable_cli=1
```

- Restart apache to apply the new configuration:

```
/opt/bitnami/ctlscript.sh restart apache
```

How to enable OpenSSL extension?

On Linux (including virtual machines and cloud images) and Mac OS X the openssl module for PHP is already enabled and no additional configuration is required.

On Windows you can enable it editing the php.ini file located in your installation directory in the php folder and uncommenting the following line:

```
extension=php_openssl.dll
```

Additionally on Windows, you may also want to add the environment variable OPENSSL_CONF so PHP finds the openssl.cnf file. In your installation the openssl.cnf file can be found in *<installdir>/apache2/conf/openssl.cnf*.

More information can be found in the official PHP documentation at <http://www.php.net/manual/en/openssl.installation.php>

How to install SSH2 for PHP?

You can use SSH2 in your PHP apps. To install and enable it, follow this steps

First of all, install the libssl-dev package

```
sudo apt-get update
sudo apt-get install libssl-dev
```

Now, download and compile libssh2 and ssh2

```
cd /tmp
wget http://www.libssh2.org/download/libssh2-1.4.3.tar.gz
tar vxzf libssh2-1.4.3.tar.gz
cd libssh2-1.4.3
./configure
make
make install
cd /tmp
wget http://pecl.php.net/get/ssh2-0.12.tgz
tar vxzf ssh2-0.12
cd ssh2-0.12
phpize
./configure --with-ssh2
make
make install
```

Finally, copy the ssh2.so file into the shared extensions folder: /opt/bitnami/php/lib/php/extensions/ and modify the php.ini file (/opt/bitnami/php/etc/php.ini) to add the following line:

```
...  
extension=ssh2.so  
...
```

Restart the server and you will now have the ability to run ssh2 commands.

Thanks to [@Chris_James](#) for posting it!

How to enable Imagick extension?

Imagick module is installed in the Bitnami PHP stacks, but is not enabled by default. To enable it, you just need to uncomment or add the following line to the php.ini file (your_installation_directory/php/etc/php.ini in Unix systems, your_installation_directory/php/php.ini in Windows systems), before the curl section

```
...  
extension=imagick.so  
...
```

Then, restart the Apache server or the php-fpm service (if available) to include the new configuration

```
/opt/bitnami/ctlscript.sh restart apache  
/opt/bitnami/ctlscript.sh restart php-fpm
```

Note: For native installers you will have to replace /opt/bitnami for the installation directory where you have installed the stack.

How to enable XCache extension?

XCache module is installed in the Bitnami PHP stacks, but is not enabled by default. To enable it, you just need to add the following lines to the php.ini file (your_installation_directory/php/etc/php.ini in Unix systems, your_installation_directory/php/php.ini in Windows systems), before the curl section

```
...  
[xcache-common]  
extension=xcache.so  
...
```

Then, restart the Apache server to include the new configuration

```
/opt/bitnami/ctlscript.sh restart apache
```

Note: For native installers you will have to replace `/opt/bitnami` for the installation directory where you have installed the stack.

How to install XCache admin?

Once you installed XCache, you can also add the administration page if you want to check more information. It is not included by default, so you need to do all the configuration manually.

First, download the proper tarball containing the XCache Administration Page for the XCache version included in the Bitnami Stack. When this wiki entry was written, the bundled XCache version was 3.1.0. You can find other versions in <http://xcache.lighttpd.net>. To check your current XCache version, you can use `php -v`

```
/opt/bitnami/php/bin/php -v
```

and you will see something similar to this

```
PHP 5.4.32 (cli) (built: Aug 25 2014 21:09:50)  
Copyright (c) 1997-2014 The PHP Group  
Zend Engine v2.4.0, Copyright (c) 1998-2014 Zend Technologies  
    with XCache v3.1.0, Copyright (c) 2005-2013, by m0o  
    with XCache Cacher v3.1.0, Copyright (c) 2005-2013, by m0o
```

Once downloaded, extract the tarball content, and you will find an `htdocs` folder. That's what you have to use to enable XCache Admin Page. But we need some steps before doing it.

1. Generate your auth password

Admin-page is protected by `http-auth`, it is disabled if you haven't yet set a password. The user/pass of `http-auth` can be set in `php.ini`. But

you have to md5(your password) first. You can use for instance this phpscript

```
<?php
echo md5("password");
?>
```

It will return your encrypted password. For instance, we used bitnami as password and returned 6a4dc9133d5f3b6d9fff778aff361961.

Now, open your php.ini file and add these lines to include your user and password.

```
xcache.admin.user = "user"
xcache.admin.pass = "6a4dc9133d5f3b6d9fff778aff361961"
```

Now, you have to create a new xcache folder and add there the Apache configuration files for it. You can learn how to create your own Bitnami Custom Application in [this wiki entry](#)

Once it is done, you should be able to access your XCache Administration Page through your browser.

How to configure NetBeans with Bitnami LAMP, MAMP or WAMP stacks?

Before starting it, you should have installed Bitnami [LAMP](#), [MAMP](#) or [WAMP](#) in your computer and also the [NetBeans IDE](#) with PHP module.

- Create your first project in NetBeans: File -> New Project
- Select the PHP category and start a new PHP application project.
- Save the source files in the "/Applications/mampstack-version/apache2/htdocs" folder on OS X or "C:\Bitnami\wampstack-version\apache2\htdocs" on Windows or "/home/user/lampstack-version/apache2/htdocs" on Linux.

netbeans-project-php.png

- Now you can select how to run your project. You can select a Local Web Site so it is only necessary to set the URL. In This case it could be different depending on your platform:

Linux and OS X: <http://localhost:8080>

Windows: <http://localhost>

[netbeans-project-php2.png](#)

- It is also possible to run PHP scripts from command line. You can set the path to the PHP binary file:

OS X: /Applications/mampstack-version/php/bin/php

Linux: /home/user/lampstack-version/php/bin/php

Windows: C:\Bitnami\wampstack-version\php\php\php.exe

[netbeans-project-php3.png](#)

- Once you started a new project you can start debugging your application from NetBeans interface. If you want to configure an existing project with Bitnami LAMP, MAMP or WAMP, you can also modify the project settings.

How to modify the PHP memory settings?

To modify the assigned memory value to PHP, you will need to modify the *memory_limit* value in */opt/bitnami/php/etc/php.ini* :

```
memory_limit = 128M
```

Also, note that your application might use *.htaccess* to configure the PHP memory limit. To know it you can run this command:

```
cd /opt/bitnami/apps  
grep -R "php_value memory_limit" .
```