Bitnami Documentation Pages (/) > General (/general/) > Components (/general/components/) > Apache (/general/components/apache/)

Bitnami Documentation For Apache

Apache is a popular open source Web server. It is a project of the Apache Software Foundation.

How To Check Which Apache Modules Are Installed?

To check which Apache modules are included in your Bitnami stack, execute the command below at the server console:

```
$ sudo /opt/bitnami/apache2/bin/apachectl -M
```

How To Check Your Certificate And Key?

If you get an error like this in the Apache error log file, it is because an incorrect certificate or key is in use.

```
[Mon May 12 15:37:46.891294 201X] [ssl:emerg] [pid 15450] AH02565: Certificate and private key example.com:443:0 from /op
t/bitnami/apps/your_app/conf/certs/server.crt and /opt/bitnami/apps/your_app/conf/certs/server.key do not match
```

- Verify that the current key matches the certificate file with the following commands. Note that the "Modulus" section in the key and certificate must match.
- Check your certificate:

  ```
  $ openssl x509 -noout -text -in server.crt -modulus | grep Modulus
    Modulus=D6E23C2E6140707EA63F3250...
  ```

- Check your key:

  ```
  $ openssl rsa -noout -text -in server.key -modulus | grep Modulus
    Modulus=D6E23C2E6140707EA63F3250...
  ```

In case of a mismatch, the wrong key is in use for the certificate and so Apache will not start until the issue is resolved.

How To Configure Your Web Application To Use A Virtual Host?

Understand Virtual Host Configuration Files

Recent versions of Bitnami apps ship three configuration files in the */opt/bitnami/apps/myapp/conf/* directory: *httpd-app.conf*, *httpd-prefix.conf* and *httpd-vhosts.conf*.

- The *httpd-app.conf* file is the main configuration file for the application. It could have different content depending on the application:

```
<Directory "/opt/bitnami/apps/myapp/htdocs">
    Options +MultiViews
    AllowOverride None
    <IfVersion < 2.3 >
    Order allow,deny
    Allow from all
    </IfVersion>
    <IfVersion >= 2.3>
    Require all granted
    </IfVersion>
</Directory>

Include /opt/bitnami/apps/myapp/conf/htaccess.conf
```

For security and performance reasons, it is advisable to not set *AllowOverride All* to anything other than *None* (refer to this Apache note (http://httpd.apache.org/docs/current/howto/htaccess.html) for more information). Bitnami applications store this configuration in the */opt/bitnami/apps/myapp/conf/htaccess.conf* file (more information).

- The *httpd-prefix.conf* file ships the default configuration for the applications in "prefix" mode, such that the application can be accessed at (for example) *http://example.com/myapp*.

```
Alias /myapp/ "/opt/bitnami/apps/myapp/htdocs/"
Alias /myapp "/opt/bitnami/apps/myapp/htdocs"

Include "/opt/bitnami/apps/myapp/conf/httpd-app.conf"
```

- The *httpd-vhosts.conf* file contains the default configuration for virtual hosts, for applications to be accessed at (for example) *http://myapp.example.com*.

```
<VirtualHost *:8080>
  ServerName myapp.example.com
  DocumentRoot "/opt/bitnami/apps/myapp/htdocs"
  Include "/opt/bitnami/apps/myapp/conf/httpd-app.conf"
</VirtualHost>

<VirtualHost *:8444>
  ServerName myapp.example.com
  DocumentRoot "/opt/bitnami/apps/myapp/htdocs"
  SSLEngine on
  SSLCertificateFile "/opt/bitnami/apps/myapp/conf/certs/server.crt"
  SSLCertificateKeyFile "/opt/bitnami/apps/myapp/conf/certs/server.key"
  Include "/opt/bitnami/apps/myapp/conf/httpd-app.conf"
</VirtualHost>
```

Define Virtual Host Configuration

To configure your application to use a virtual host instead of the prefix URL, make these changes:

- Delete the following line in the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf* file:

```
Include "/opt/bitnami/apps/myapp/conf/httpd-prefix.conf"
```

- Add a new link in the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file:

```
Include "/opt/bitnami/apps/myapp/conf/httpd-vhosts.conf"
```

- Some applications require further changes in configuration files or the database. Please check the exact changes in the application's documentation.

NOTE: After modifying the Apache configuration files, restart Apache to apply the changes.

How To Configure Multiple SSL Domains On The Same IP Address?

There is an extension to the SSL protocol called "Server Name Indication". It allows you to use only one IP address for several SSL-protected sites. The only drawback is that some older web browsers do not support it. The example Apache configuration is shown below:

```
NameVirtualHost *:80

<VirtualHost *:80>
ServerName my-wordpress.example.com
DocumentRoot "/opt/bitnami/apps/wordpress/htdocs"
</VirtualHost>
<VirtualHost *:80>
ServerName my-sugarcrm.example.com
DocumentRoot "/opt/bitnami/apps/sugarcrm/htdocs"
</VirtualHost>

Listen 443
NameVirtualHost *:443

<VirtualHost *:443>
SSLEngine on
DocumentRoot "/opt/bitnami/apps/wordpress/htdocs"
ServerName my-wordpress.example.com
SSLCertificateFile "/opt/bitnami/apache2/conf/my-wordpress.crt"
SSLCertificateKeyFile "/opt/bitnami/apache2/conf/my-wordpress.key"
</VIrtualHost>

<VirtualHost *:443>
SSLEngine on
DocumentRoot "/opt/bitnami/apps/sugarcrm/htdocs"
ServerName my-sugarcrm.example.com
SSLCertificateFile "/opt/bitnami/apache2/conf/my-sugarcrm.crt"
SSLCertificateKeyFile "/opt/bitnami/apache2/conf/my-sugarcrm.key"
</VirtualHost>
```

You can learn more at the following pages:

- http://wiki.apache.org/httpd/NameBasedSSLVHostsWithSNI (http://wiki.apache.org/httpd/NameBasedSSLVHostsWithSNI)
- http://serverfault.com/questions/109800/multiple-ssl-domains-on-the-same-ip-address-and-same-port (http://serverfault.com/questions/109800/multiple-ssl-domains-on-the-same-ip-address-and-same-port)

How To Change The Apache Port?

HTTP Port

Under the default configuration, Apache will wait for requests on port 80. Change that by editing the *httpd.conf* file and modifying the value specified in the *Port* directive. For example:

```
Listen 8080

ServerName localhost:8080
```

Also change the port in */opt/bitnami/apache2/conf/bitnami/bitnami.conf* in the *VirtualHost* directive:

```
<VirtualHost _default_:8080>
```

Restart the Apache server for the change to take effect.

HTTPS Port

Apache waits for HTTPS requests on port 443. Change that by editing the */opt/bitnami/apache2/conf/bitnami/bitnami.conf* file and modifying the value specified in the *Port* directive. For example:

```
Listen 8443

<VirtualHost _default_:8443>
```

Restart the Apache server for the change to take effect.

How To Create An SSL Certificate?

OpenSSL is required to create an SSL certificate. A certificate request can then be sent to a certificate authority (CA) to get it signed into a certificate, or if you have your own certificate authority, you may sign it yourself, or you can use a self-signed certificate (because you just want a test certificate or because you are setting up your own CA).

Follow the steps below:

- Create your private key (if you haven't created it already):

  ```
  $ sudo openssl genrsa -out /opt/bitnami/apache2/conf/server.key 2048
  ```

- Create a certificate:

  ```
  $ sudo openssl req -new -key /opt/bitnami/apache2/conf/server.key -out /opt/bitnami/apache2/conf/cert.csr
  ```

  IMPORTANT: Enter the server domain name when the above command asks for the "Common Name".

- Send *cert.csr* to the certificate authority. When the certificate authority completes their checks (and probably received payment from you), they will hand over your new certificate to you.
- Until the certificate is received, create a temporary self-signed certificate:

  ```
  $ sudo openssl x509 -in /opt/bitnami/apache2/conf/cert.csr -out /opt/bitnami/apache2/conf/server.crt -req -signkey
   /opt/bitnami/apache2/conf/server.key -days 365
  ```

- Back up your private key in a safe location after generating a password-protected version as follows:

  ```
  $ sudo openssl rsa -des3 -in /opt/bitnami/apache2/conf/server.key -out privkey.pem
  ```

  Note that if you use this encrypted key in the Apache configuration file, it will be necessary to enter the password manually every time Apache starts. Regenerate the key without password protection from this file as follows:

  ```
  $ sudo openssl rsa -in privkey.pem -out /opt/bitnami/apache2/conf/server.key
  ```

Find more information about certificates at http://www.openssl.org (http://www.openssl.org).

How To Create A Virtual Host?

Using a Virtual Host allows you to access an application at (for example) *http://SERVER-IP/* or *http://APPNAME.SERVER-IP* instead of *http://SERVER-IP/APPNAME*.

This example shows how to configure WordPress to be accessible from *http://DOMAIN* (replace the DOMAIN placeholder with the correct domain name for your virtual host). Follow these steps:

- Comment out the line that includes the prefix configuration file in the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf* file:

  ```
    # Include "/opt/bitnami/apps/wordpress/conf/httpd-prefix.conf"
  ```

- Include the virtual host configuration file for WordPress in the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file:

```
Include "/opt/bitnami/apps/wordpress/conf/httpd-vhosts.conf"
```

- Edit the */opt/bitnami/apps/wordpress/conf/httpd-vhosts.conf* file and replace the placeholder domain within the *ServerName* and *ServerAlias* directives with the correct domain name.
- Update the URL in the application if necessary. For WordPress, you will need to update the WordPress database using the following command. Remember to replace the DOMAIN placeholder with the correct domain name for your virtual host.

```
$ sudo mysql -u root -p -e "USE bitnami_wordpress; UPDATE wp_options SET option_value='http://DOMAIN' WHERE option_name='siteurl' OR option_name='home';"
```

- Restart the Apache server:

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

How To Debug Apache Errors?

Once Apache starts, it will create two log files at */opt/bitnami/apache2/logs/access_log* and */opt/bitnami/apache2/logs/error_log* respectively.

- The *access_log* file is used to track client requests. When a client requests a document from the server, Apache records several parameters associated with the request in this file, such as: the IP address of the client, the document requested, the HTTP status code, and the current time.
- The *error_log* file is used to record important events. This file includes error messages, startup messages, and any other significant events in the life cycle of the server. This is the first place to look when you run into a problem when using Apache.

If no error is found, you will see a message similar to:

```
Syntax OK
```

How To Deny Connections From Bots/Attackers?

Sometimes, if you are experiencing poor performance, it is because you are being attacked by Internet bots. The reason for these attacks is that they are trying to find a security bug in your application code or in the software itself.

An example of a bot attack is attempting to check if the *php.cgi* binary is disabled. As this is disabled by default, attackers won't be able to exploit your system, but you will have hundreds or even thousands of connections from the same IP address (or even different IP addresses) trying to "check" every few hours if those binaries or scripts are available.

Our stacks and cloud images come with the latest versions of their components but, even though you are safe from those attacks, your server could experience poor performance because of the traffic they generate.

To know if you are being attacked, run the command below:

```
$ cd /opt/bitnami/apache2/logs/
$ tail -n 10000 access_log | awk '{print $1}'| sort| uniq -c| sort -nr| head -n 10
```

This will show you the number of times that an IP address connected to your Web server. If you see that some IP addresses have many more connections than others, run the following command (remember to modify ATTACKER_IP with the correct IP):

```
$ cd /opt/bitnami/apache2/logs/
$ grep "ATTACKER_IP" access_log
```

If you see that the IP address is always attempting to connect to the same location, if it is a URL that you don't know, or if it is trying to run binaries or scripts directly, it is likely that IP address is a bot.

Examples of log messages for this scenario are:

```
[Mon Dec 08 07:01:52 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php-cgi
[Mon Dec 08 07:01:52 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php.cgi
[Mon Dec 08 07:01:53 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php4
[Mon Dec 08 19:01:51 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php
[Mon Dec 08 19:01:51 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php5
[Mon Dec 08 19:01:52 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php-cgi
[Mon Dec 08 19:01:52 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php.cgi
[Mon Dec 08 19:01:52 2014] [error] [client 143.107.202.68] script not found or unable to stat: /opt/bitnami/apache2/cgi-bin/php4
```

This shows that an attacker with IP address 143.107.202.68 is trying to find the PHP CGI scripts, and all these connections are taking place within the same second.

To deny connections to these attackers, the easiest way is with your Apache configuration file. Follow these steps:

- Edit the file at */opt/bitnami/apps/APPNAME/conf/httpd-app.conf*. The example below shows how to reject the 1.2.3.4 IP address in WordPress:

  ```
  <Directory /opt/bitnami/apps/wordpress/htdocs>
  deny from 1.2.3.4
  ...
  </Directory>
  ```

  To deny access to more than one IP, use the example below:

  ```
  <Directory /opt/bitnami/apps/wordpress/htdocs>
  deny from 1.2.3.4
  deny from 5.6.7.8
  deny from 9.10.11.12
  ...
  </Directory>
  ```

- Check if your changes are okay by executing the following command:

  ```
  $ apachectl -t
  ```

- Restart the Apache web server:

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

To further protect your website, consider installing the *mod_evasive* module.

How To Disable The Cache In The Server?

If you are developing on top of an AMP Stack or customizing any Bitnami Stack, your files (like JavaScript files) may be cached by the server and even you modify them your changes will not appear to be applied.

In order to disable the cache in the server and let the files be served each time, disable PageSpeed for Apache and OPCache, enabled by default in PHP.

To disable PageSpeed, comment out the following lines in your *httpd.conf* (*/opt/bitnami/apache2/conf/httpd.conf*)

```
#Include conf/pagespeed.conf

#Include conf/pagespeed_libraries.conf
```

To disable OPCache, change *opcache.enable* in your *php.ini* file and set it to 0 (*/opt/bitnami/php/etc/php.ini*)

How To Enable HTTPS Support With SSL Certificates?

> NOTE: The steps below assume that you are using a custom domain name and that you have already configured the custom domain name to point to your cloud server.

Bitnami images come with SSL support already pre-configured and with a dummy certificate in place. Although this dummy certificate is fine for testing and development purposes, you will usually want to use a valid SSL certificate for production use. You can either generate this on your own (explained here) or you can purchase one from a commercial certificate authority.

Once you obtain the certificate and certificate key files, you will need to update your server to use them. Follow these steps to activate SSL support:

- Use the table below to identify the correct locations for your certificate and configuration files.

Variable	Value
Current application URL	https://[custom-domain]/
	Example: https://my-domain.com/ or https://my-domain.com/appname
Apache configuration file	/opt/bitnami/apache2/conf/bitnami/bitnami.conf
Certificate file	/opt/bitnami/apache2/conf/server.crt
Certificate key file	/opt/bitnami/apache2/conf/server.key
CA certificate bundle file (if present)	/opt/bitnami/apache2/conf/server-ca.crt

- Copy your SSL certificate and certificate key file to the specified locations.

> NOTE: If you use different names for your certificate and key files, you should reconfigure the *SSLCertificateFile* and *SSLCertificateKeyFile* directives in the corresponding Apache configuration file to reflect the correct file names.

- If your certificate authority has also provided you with a PEM-encoded Certificate Authority (CA) bundle, you must copy it to the correct location in the previous table. Then, modify the Apache configuration file to include the following line below the *SSLCertificateKeyFile* directive. Choose the correct directive based on your scenario and Apache version:

Variable	Value
Apache configuration file	/opt/bitnami/apache2/conf/bitnami/bitnami.conf
Directive to include (Apache v2.4.8+)	SSLCACertificateFile "/opt/bitnami/apache2/conf/server-ca.crt"
Directive to include (Apache < v2.4.8)	SSLCertificateChainFile "/opt/bitnami/apache2/conf/server-ca.crt"

> NOTE: If you use a different name for your CA certificate bundle, you should reconfigure the *SSLCertificateChainFile* or *SSLCACertificateFile* directives in the corresponding Apache configuration file to reflect the correct file name.

- Once you have copied all the server certificate files, you may make them readable by the root user only with the following commands:

```
$ sudo chown root:root /opt/bitnami/apache2/conf/server*
$ sudo chmod 600 /opt/bitnami/apache2/conf/server*
```

- Open port 443 in the server firewall. Refer to the FAQ (/general/faq#how-to-open-the-server-ports-for-remote-access) for more information.
- Restart the Apache server.

You should now be able to access your application using an HTTPS URL.

How To Enable LDAP Module In Apache?

Bitnami stacks already ship the LDAP module installed in Apache but it is not enabled by default. To enable this module, follow these steps:

- Enable the LDAP module. Edit the main Apache configuration file located at */opt/bitnami/apache2/conf/httpd.conf*. Uncomment the *mod_authnz_ldap* line and add the *mod_ldap* line at the end of the *LoadModule* section:

```
...
LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
...
LoadModule ldap_module modules/mod_ldap.so
```

- Restart Apache server and check it is already enabled:

```
$ sudo /opt/bitnami/ctlscript.sh restart apache
$ /opt/bitnami/apache2/bin/apachectl -M | grep ldap
  ...
  authnz_ldap_module (shared)
  ldap_module (shared)
  ...
```

How To Add The *Mod_evasive* Module In Apache?

Follow these steps:

- Download the latest version:

```
$ wget http://www.zdziarski.com/blog/wp-content/uploads/2010/02/mod_evasive_1.10.1.tar.gz
```

- Extract the content:

```
$ tar zxvf mod_evasive_1.10.1.tar.gz
```

- Build, configure and install the module:

```
$ cd mod_evasive
$ cp mod_evasive{20,24}.c
$ sed s/remote_ip/client_ip/g -i mod_evasive24.c
$ sudo apxs -i -a -c mod_evasive24.c
```

- Update the Apache module configuration:

```
$ sudo sed 's@Include "/opt/bitnami/apache2/conf/bitnami/bitnami.conf"@Include "/opt/bitnami/apache2/conf/bitnami/bitnami.conf"\nInclude "/opt/bitnami/apache2/conf/modevasion.conf"@' -i /opt/bitnami/apache2/conf/httpd.conf
$ sudo tee /opt/bitnami/apache2/conf/modevasion.conf <<EOF
 #increases size of hash table. Good, but uses more RAM."
 DOSHashTableSize    3097"
 #Interval, in seconds, of the page interval."
 DOSPageInterval     1"
 #Interval, in seconds, of the site interval."
 DOSSiteInterval     1"
 #period, in seconds, a client is blocked.  The counter is reset to 0 with every access within this interval."
 DOSBlockingPeriod   10"
 #threshold of requests per page, per page interval.  If hit == block."
 DOSPageCount        2"
 #threshold of requests for any object by the same ip, on the same listener, per site interval."
 DOSSiteCount        50"
 #locking mechanism prevents repeated calls.  email can be sent when host is blocked (leverages the following by default "/bin/mail -t %s")"
 DOSEmailNotify      mbrown@domainy.com"
 #locking mechanism prevents repeated calls.  A command can be executed when a host is blocked.  %s is the host IP."
 #DOSSystemCommand    \"su - someuser -c \'/sbin/... %s ...\'\""
 #DOSLogDir           \"/var/lock/mod_evasive\""
 #whitelist an IP., leverage wildcards, not CIDR, like 127.0.0.*"
 #DOSWhiteList 127.0.0.1"
 EOF
```

- Restart Apache:

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

How To Add The *Mod_proxy_html* Module In Apache?

Follow these steps:

- Download the latest version:

```
$ wget http://apache.webthing.com/mod_proxy_html/mod_proxy_html.tar.bz2
```

- Extract the content and install the module:

```
$ tar -jxf mod_proxy_html.tar.bz2
$ cd mod_proxy_html/
$ sudo apxs -c -I /opt/bitnami/common/include/libxml2 -I. -i mod_proxy_html.c
$ sudo chmod 755 /opt/bitnami/apache2/modules/mod_proxy_html.so
$ sudo apxs -c -I /opt/bitnami/common/include/libxml2 -I. -i mod_xml2enc.
$ sudo chmod 755 /opt/bitnami/apache2/modules/mod_xml2enc.so
```

- Enable the module by including the lines below in the */opt/bitnami/apache2/conf/httpd.conf* configuration file:

```
 LoadFile /opt/bitnami/common/lib/libxml2.so
 LoadModule proxy_html_module modules/mod_proxy_html.so
 LoadModule xml2enc_module modules/mod_xml2enc.so
```

How To Add The *Mod_rpaf* Module In Apache?

Follow these steps:

- Download the latest version:

```
$ wget https://github.com/gnif/mod_rpaf/archive/stable.zip
```

- Extract the contents and install the module:

```
$ unzip stable.zip
$ cd mod_rpaf-stable
$ sudo make
$ sudo make install
```

- Check that the *mod_rpaf.so* file exists in the */opt/bitnami/apache2/modules* directory:

```
$ ll /opt/bitnami/apache2/modules/mod_rpaf.so
```

- Load and configure the module. A configuration example follows; this can be added to the Apache configuration file at /opt/bitnami/apache2/conf/httpd.conf.

```
LoadModule             rpaf_module modules/mod_rpaf.so
RPAF_Enable            On
RPAF_ProxyIPs          127.0.0.1 10.0.0.0/24
RPAF_SetHostName       On
RPAF_SetHTTPS          On
RPAF_SetPort           On
RPAF_ForbidIfNotProxy  Off
```

- Restart Apache to reload the new configuration:

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

How To Enable *Mod_security* In Apache?

Bitnami stacks already ship the *mod_security2* module installed in Apache but it is not enabled by default. To enable this module, follow these steps:

- Enable the *mod_security2* and *mod_unique_id* modules in Apache. Edit the main Apache configuration file and uncomment the *unique_id_module* and add the *mod_security* line at the end of the *LoadModule* section:

```
...
LoadModule unique_id_module modules/mod_unique_id.so
...
LoadModule security2_module modules/mod_security2.so
```

- Add the default configuration file for *mod_security* at the end of the Apache configuration file:

```
Include "/opt/bitnami/apache2/conf/modsecurity.conf"
```

- Restart Apache server and check it is already enabled:

```
$ sudo /opt/bitnami/ctlscript.sh restart apache
$ tail /opt/bitnami/apache2/logs/error_log
```

```
...
 [Thu Jan 30 18:42:14.004246 2014] [:notice] [pid 1127] ModSecurity for Apache/2.6.7 (http://www.modsecurity.org/)
configured.
...
```

How To Add The *Mod_xsendfile* Module In Apache?

Bitnami LAMP/MAMP/WAMP stacks b5.4.13-2 and later include the *mod_xsendfile* module. To enable this module, add the following line in the Apache configuration file:

```
LoadModule xsendfile_module modules/mod_xsendfile.so
```

If you are using an older version, it is easy to install this module into your existing Apache server. Follow these steps:

- Download the latest version:

```
$ wget https://tn123.org/mod_xsendfile/mod_xsendfile-0.12.tar.gz
```

- Extract the content and install the module:

```
$ tar -xzvf mod_xsendfile-0.12.tar.gz
$ cd mod_xsendfile-0.12
$ sudo /opt/bitnami/apache2/bin/apxs -aci mod_xsendfile.c
```

If everything goes well, the module will be installed to */opt/bitnami/apache2/modules/mod_xsendfile.so*. Check the mod_xsenfile configuration page (https://tn123.org/mod_xsendfile/) to find out how to configure this module for your application.

How To Configure Apache With Phusion Passenger?

To configure Apache with Phusion Passenger, refer to this page (/general/components/passenger).

How To Force HTTPS Redirection For An Application?

Add the following to the top of the */opt/bitnami/apps/APPNAME/conf/httpd-prefix.conf* file:

```
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [R,L]
```

After modifying the Apache configuration files:

- Open port 443 in the server firewall. Refer to the FAQ (/general/faq#how-to-open-the-server-ports-for-remote-access) for more information.
- Restart Apache to apply the changes.

How To Force HTTPS For All Applications?

This depends on your current Apache configuration, but in most cases it should be enough to add the following lines in the default Apache virtual host configuration file at */opt/bitnami/apache2/conf/bitnami/bitnami.conf*, inside the default *VirtualHost* directive:

```
<VirtualHost _default_:80>
  DocumentRoot "/opt/bitnami/apache2/htdocs"
  RewriteEngine On
  RewriteCond %{HTTPS} !=on
  RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [R,L]
  ...
</VirtualHost>
```

After modifying the Apache configuration files:

- Open port 443 in the server firewall. Refer to the FAQ (/general/faq#how-to-open-the-server-ports-for-remote-access) for more information.
- Restart Apache to apply the changes.

How To Install A Let's Encrypt Certificate In Your Web Server?

To learn more about this topic, read our guide on generating and installing Let's Encrypt certificates for Bitnami applications (/general/how-to/generate-install-lets-encrypt-ssl).

How To Publish My Web Page?

If you already have a Web page and you want to serve its content with Apache, copy your file to the default document root directory at */opt/bitnami/apache2/htdocs/*.

How To Redirect Www.Myapp.Example.Com (Or Other Domains) To My Server?

- Add a *ServerAlias* in the *httpd-vhosts.conf* file for your application. This option is designed to specify alternate names for a host and is used when matching requests. Here's an example:

```
<VirtualHost *:80>
ServerName app.example.com
ServerAlias www.app.example.com app.example.org www.app.example.uk.org
...

<VirtualHost *:443>
ServerName app.example.com
ServerAlias www.app.example.com app.example.org www.app.example.uk.org
```

- Check that the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file includes the *httpd-vhosts.conf* file for your application. It should include a line like the one below.

```
Include "/opt/bitnami/apps/APPNAME/conf/httpd-vhosts.conf"
```

If it does not, add the line above to the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file, replacing the APPNAME placeholder with the correct directory name for your application.

NOTE: After modifying the Apache configuration files, restart Apache to apply the changes.

How To Redirect Www.Myapp.Example.Com To Myapp.Example.Com?

This redirection is an SEO "best practice".

- Add the following in the *httpd-vhosts.conf* file for your application.

```
<VirtualHost *:80>
ServerName app.example.com
ServerAlias www.app.example.com
RewriteEngine On
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ http://%1$1 [R=permanent,L]
...

<VirtualHost *:443>
ServerName app.example.com
ServerAlias www.app.example.com
RewriteEngine On
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ https://%1$1 [R=permanent,L]
...
```

- Check that the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file includes the *httpd-vhosts.conf* file for your application. It should include a line like the one below.

```
Include "/opt/bitnami/apps/APPNAME/conf/httpd-vhosts.conf"
```

If it does not, add the line above to the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file, replacing the APPNAME placeholder with the correct directory name for your application.

NOTE: After modifying the Apache configuration files, restart Apache to apply the changes.

How To Redirect Myapp.Example.Com To Www.Myapp.Example.Com?

- Add the following in the *httpd-vhosts.conf* file for your application. Or, to apply this redirection by default for all applications installed, add it to the default *VirtualHost* in the */opt/bitnami/apache2/conf/bitnami.conf* file.

```
<VirtualHost *:80>
  ServerName app.example.com
  ServerAlias www.app.example.com
  RewriteEngine On
  RewriteCond %{HTTP_HOST} !^www\. [NC]
  RewriteRule ^(.*)$ http://www.%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
 ...

<VirtualHost *:443>
  ServerName app.example.com
  ServerAlias www.app.example.com
  RewriteEngine On
  RewriteCond %{HTTP_HOST} !^www\. [NC]
  RewriteRule ^(.*)$ https://www.%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
 ...
```

- If you used the *httpd-vhosts.conf* file for the application, check that the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file includes the *httpd-vhosts.conf* file for your application. It should include a line like the one below.

```
Include "/opt/bitnami/apps/APPNAME/conf/httpd-vhosts.conf"
```

  If it does not, add the line above to the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file, replacing the APPNAME placeholder with the correct directory name for your application.

NOTE: After modifying the Apache configuration files, restart Apache to apply the changes.

Troubleshooting

Why Can't I Start The Apache Server?

- Check the Apache error log file

  Check the Apache error log file at */opt/bitnami/apache2/logs/error_log* for information about why the error occurred.

- Check if another process is listening to that port

  If another process is using that address you'll get:

```
(98)Address already in use: AH00072: make_sock: could not bind to address 0.0.0.0:port_number
no listening sockets available, shutting down
```

  To see which process is already using that port you can run the following from a command prompt:

```
$ sudo netstat -ltnp | grep :port_number
```

  In the last column you'll see the process id or process name. You can then use:

```
$ ps aux | grep process_name
```

  Look for the *pid* in the second column and you'll get more information about that process.

  In case another process is using that port, use another port or stop that process.

- Check permissions and ownership

  Check if you have permissions to bind Apache to the requested port. To bind Apache to privileged ports, start Apache as root. If you don't have permissions to bind Apache to some port, you'll see this error:

```
(13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:port_number
no listening sockets available, shutting down
```

If Apache is unable to open the configuration or the log file, check that the owner of those files is the same user account that installed Apache and that it has write permissions on logs and read permissions on the configuration file. If this is not the case, you will see these errors:

```
(13)Permission denied: AH00649: could not open transfer log file /opt/bitnami/apache2/logs/access_log.
AH00015: Unable to open logs

(13)Permission denied: AH00091: httpd: could not open error log file /opt/bitnami/apache2/logs/error_log.
AH00015: Unable to open logs

httpd: Could not open configuration file /opt/bitnami/apache2/conf/httpd.conf: Permission denied
apache config test fails, aborting
```

What Does The Message "Your Connection To This Site Is Only Partially Encrypted…" Mean?

This message appears when you enable SSL for your site but there are some resources referenced by unencrypted HTTP URLs in your page.

To check if this is the case, view the page source and check for any *http://* references. To resolve the issue, manually update your themes or templates and change any URLs to relative URLs. More specifically, instead of using *http://* in your code, use *//:*, as below:

```
<img src='//example.com/img.png'/>
```

How To Configure The Apache Server?

The main Apache configuration file is located at */opt/bitnami/apache2/conf/httpd.conf*.

In recent versions of Bitnami stacks, the */opt/bitnami/apache2/conf/bitnami/bitnami.conf* file defines which configuration for each application should be loaded by the Apache server.

By default, Bitnami applications are accessible at *http://SERVER-IP/APP*. The list of applications to load is included in the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf* file. Those applications that need to be served in a different virtual host should be included in the */opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf* file.

Which Is My Apache Version?

Check your Apache version using the *apachectl* command. Remember to execute it inside the Bitnami console:

```
$ apachectl -V
```

What Is The Apache Event MPM?

Apache supports three different Multi-Processing Modules (MPMs) that are responsible for binding to network ports on the machine, accepting requests, and dispatching children to handle the requests.

- Prefork: Each child process handles one connection at a time. This is the default mode for Apache.
- Worker: It uses multiple child processes with many threads each.
- Event: This MPM was designed to allow more requests to be served simultaneously by passing off some processing work to supporting threads and freeing up the main threads to work on new requests.

Bitnami stacks ship all the different MPMs as shared modules so it is possible to configure Apache to enable any of them.

Native installers are configured for development purposes. Apache is configured to use prefork mode. PHP applications are configured to use *mod_php*. If you use a native installer and you want to modify your configuration manually, refer to our PHP-FPM page (/general/components/php-fpm).

Virtual machines and cloud images are configured for production environments by default. Apache is configured to use the event mode and PHP applications will use the PHP-FPM server for the requests.

What Is PHP-FPM?

PHP-FPM (FastCGI Process Manager) is an alternative PHP FastCGI implementation with some additional features useful for heavily-trafficked websites. It has been bundled with PHP since v5.3.3. PHP-FPM has the ability to start workers with different environments and to manage these processes (more information (http://www.php.net/manual/en/install.fpm.php)).

The recommended configuration is to configure Apache to serve static files (images, CSS, JavaScript and more) and use PHP_FPM with Apache's *mod_proxy* module to handle PHP requests.

Why MPM Event And PHP-FPM Is Recommended For Production?

The default configuration for Apache and PHP in Linux systems is to use the prefork MPM with an embedded PHP interpreter. This is a very robust configuration but it means that Apache needs to spawn a separate process for every simultaneous request it wants to serve. Because every child process loads a PHP interpreter and associated libraries, this configuration takes a significant amount of memory. In addition to this, a whole process is tied up when waiting for requests when browsers open a persistent connection to the server (which is particularly common with AJAX-heavy web applications).

On high-traffic websites, an alternate MPM (the event MPM) is preferable because it has the ability to serve a large amount of requests while maintaining low memory usage. It does so by using threads to serve requests. It retains some of the stability of a process-based server by keeping multiple processes available, each with many threads so a thread potentially misbehaving would only affect all the other threads in the same process.

Additionally, the event MPM uses a dedicated thread to deal with kept-alive connections, and hands requests down to child threads only when a request has actually been made. That allows those threads to free back up immediately after the request is completed.

Because PHP is not thread-safe on Unix environments (in particular many of its most popular extensions), it is not possible to embed an interpreter in each event MPM process. It needs to live as a separate PHP-FPM process. Apache will serve all static resources (images, CSS, etc.) directly while PHP-FPM serves the PHP-related requests.

In the examples below we provide some benchmarks. We used two different Amazon EC2 instances (micro and small) to run our tests. Both instances had the same WordPress installation and the same memory settings. We used the Siege tool for HTTP load testing and benchmarking. We also used the webpagetest.org tool (http://www.webpagetest.org/) that allows running speed test from multiple locations using real browsers.

The test below uses the Siege tool to request the WordPress Web page and all its static files. We used 30 concurrent users over 1 minute and obtained the following results:

	Small EC2 instance with MPM prefork and *mod_php*	Small EC2 instance with MPM event and PHP-FPM
Used memory	525MB (Apache)	78MB + 200MB (max) (Apache + PHP-FPM)
Transactions	1606 hits	2480 hits
Availability	89.92 %	91.75 %
Elapsed time	59.08 secs	59.10 secs
Data transferred	12.49 MB	21.30 MB
Response time	0.89 secs	0.69 secs
Transaction rate	27.18 trans/sec	41.96 trans/sec
Throughput	0.21 MB/sec	0.36 MB/sec
Concurrency	24.28	29.11
Successful transactions	1411	2480
Failed transactions	180	223
Longest transaction	5.89	6.17

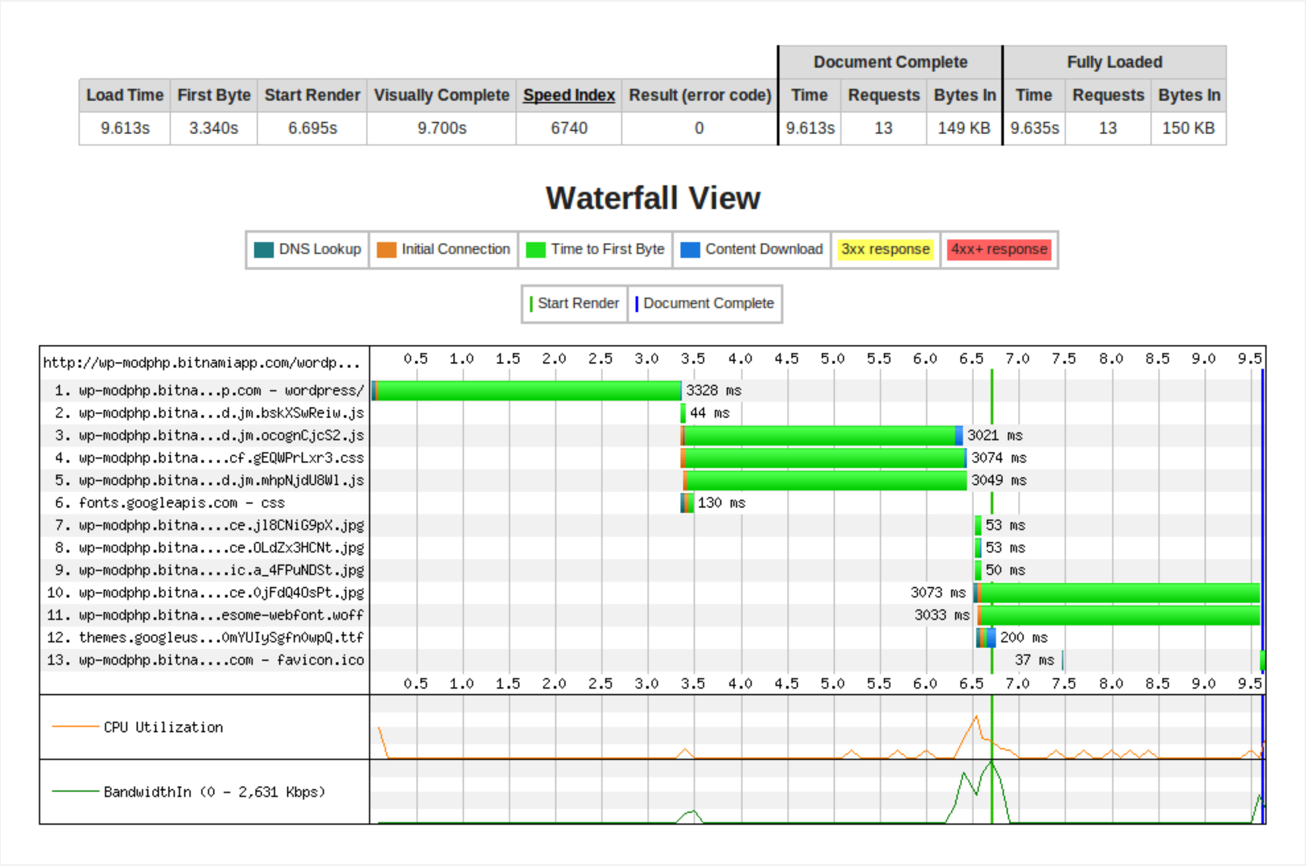
Shortest transaction	0.11	0.11

The main differences are the following:

- The used memory is much lower.
- The amount of data transferred is much higher.
- The transaction rate is higher and there are less failed transactions.

The next test shows the connection and page serving times of 9 concurrent users using EC2 micro instances with the same WordPress sample site.

Apache MPM prefork with *mod_php*:

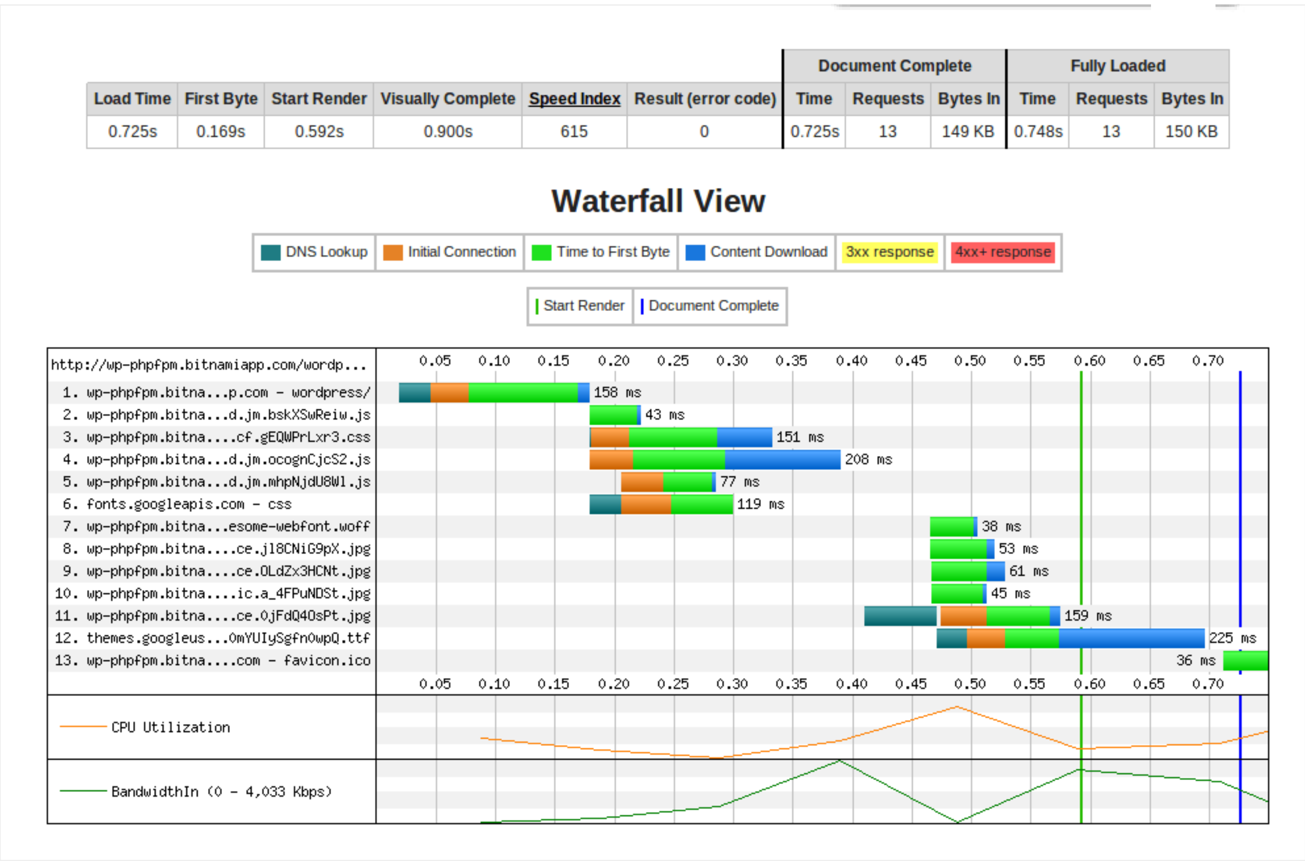


						Document Complete			Fully Loaded		
Load Time	First Byte	Start Render	Visually Complete	Speed Index	Result (error code)	Time	Requests	Bytes In	Time	Requests	Bytes In
9.613s	3.340s	6.695s	9.700s	6740	0	9.613s	13	149 KB	9.635s	13	150 KB

Waterfall View
DNS Lookup
Initial Connection
Time to First Byte
Content Download
3xx response
4xx+ response
Start Render
Document Complete
http://wp-modphp.bitnamiapp.com/wordp...
0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9.0 9.5
1. wp-modphp.bitna...p.com - wordpress/ 3328 ms
2. wp-modphp.bitna...d.jm.bskXSwReiw.js 44 ms
3. wp-modphp.bitna...d.jm.ocognCjcS2.js 3021 ms
4. wp-modphp.bitna....cf.gEQWPrLxr3.css 3074 ms
5. wp-modphp.bitna...d.jm.mhpNjdU8Wl.js 3049 ms
6. fonts.googleapis.com - css 130 ms
7. wp-modphp.bitna....ce.jl8CNiG9pX.jpg 53 ms
8. wp-modphp.bitna....ce.OLdZx3HCNt.jpg 53 ms
9. wp-modphp.bitna....ic.a_4FPuNDSt.jpg 50 ms
10. wp-modphp.bitna....ce.0jFdQ40sPt.jpg 3073 ms
11. wp-modphp.bitna...esome-webfont.woff 3033 ms
12. themes.googleus...0mYUIySgfn0wpQ.ttf 200 ms
13. wp-modphp.bitna....com - favicon.ico 37 ms
CPU Utilization
BandwidthIn (0 - 2,631 Kbps)

(/images/img/components/apache/apache-wordpress-modphp-e360b6f7.png)

Apache MPM event with PHP-FPM:



						Document Complete			Fully Loaded		
Load Time	First Byte	Start Render	Visually Complete	Speed Index	Result (error code)	Time	Requests	Bytes In	Time	Requests	Bytes In
0.725s	0.169s	0.592s	0.900s	615	0	0.725s	13	149 KB	0.748s	13	150 KB

Waterfall View

DNS Lookup
Initial Connection
Time to First Byte
Content Download
3xx response
4xx+ response
Start Render
Document Complete
http://wp-phpfpm.bitnamiapp.com/wordp...
1. wp-phpfpm.bitna...p.com - wordpress/
2. wp-phpfpm.bitna...d.jm.bskXSwReiw.js
3. wp-phpfpm.bitna....cf.gEQWPrLxr3.css
4. wp-phpfpm.bitna...d.jm.ocognCjcS2.js
5. wp-phpfpm.bitna...d.jm.mhpNjdU8Wl.js
6. fonts.googleapis.com - css
7. wp-phpfpm.bitna...esome-webfont.woff
8. wp-phpfpm.bitna....ce.jl8CNiG9pX.jpg
9. wp-phpfpm.bitna....ce.OLdZx3HCNt.jpg
10. wp-phpfpm.bitna....ic.a_4FPuNDSt.jpg
11. wp-phpfpm.bitna....ce.0jFdQ40sPt.jpg
12. themes.googleus...OmYUIySgfnOwpQ.ttf
13. wp-phpfpm.bitna....com - favicon.ico
0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70
158 ms
43 ms
151 ms
208 ms
77 ms
119 ms
38 ms
53 ms
61 ms
45 ms
159 ms
225 ms
36 ms
CPU Utilization
BandwidthIn (0 - 4,033 Kbps)

[(/images/img/components/apache/apache-wordpress-phpfpm-42477e11.png)](/images/img/components/apache/apache-wordpress-phpfpm-42477e11.png)

The main differences are the following:

- The load time is much lower in the second case. It is also important that the load time in the first case could be very different for different tests, as it depends on the number of free child processes.
- The "time to first byte" is much lower in the second case.

The previous test only had 11 static files but in a production PHP application, it is usual to have more and bigger static files. The following example shows a simple static HTML page with 240 small PNG images. We used the Siege tool with 30 concurrent users over 30 seconds.

	Small EC2 instance with MPM prefork	Small EC2 instance with MPM event
Used Memory	182MB	66MB
Transactions	2148 hits	3568 hits
Availability	98.90 %	100.00 %
Elapsed time	29.38 secs	29.23 secs
Data transferred	1.10 MB	1.82 MB
Response time	0.25 secs	0.24 secs
Transaction rate	73.11 trans/sec	122.07 trans/sec
Throughput	0.04 MB/sec	0.06 MB/sec
Concurrency	18.32	29.41
Successful transactions	2148	3568
Failed transactions	24	0
Longest transaction	10.55	0.94
Shortest transaction	0.11	0.11

The main differences are the following:

- The used memory is much lower in the second case.
- The Transaction rate is much higher in the second case.
- The Longest transaction is similar to the Shortest transaction.

In conclusion, the Apache event MPM increases the performance of your Apache server, allowing it to serve more requests with less memory. If you want to deploy a PHP application, use PHP-FPM to handle PHP requests. This is highly recommended for running applications on servers with limited memory, such as cloud instances with 512 MB or 1 GB RAM. Bitnami stacks use this configuration by default for virtual machines and cloud images.

How To Configure *.Htaccess* Files?

One of our main goals is to configure Bitnami applications in the most secure way. For this reason, we moved the configuration in the *.htaccess* files to the main application configuration files and set the *AllowOverride* option to *None* by default.

> NOTE: The Apache Software Foundation also recommends this configuration (http://httpd.apache.org/docs/current/mod/core.html#allowoverride). To quote: "For security and performance reasons, do not set AllowOverride to anything other than None in your block. Instead, find (or create) the block that refers to the directory where you're actually planning to place a .htaccess file."

The content of the *.htaccess* files have been moved to the */opt/bitnami/apps/APPNAME/conf/htaccess.conf* file. For example, the Bitnami MediaWiki application uses the following configuration files:

- The */opt/bitnami/apps/mediawiki/conf/httpd-app.conf* file is the main application configuration file (previous versions called it *mediawiki.conf*). It also sources the *htaccess.conf* file.

```
<Directory "/opt/bitnami/apps/mediawiki/htdocs">
    Options +MultiViews
    AllowOverride None
    <IfVersion < 2.3 >
    Order allow,deny
    Allow from all
    </IfVersion>
    <IfVersion >= 2.3>
    Require all granted
    </IfVersion>
</Directory>
Include "/opt/bitnami/apps/mediawiki/conf/htaccess.conf"
```

- The */opt/bitnami/apps/mediawiki/conf/htaccess.conf* file ships the content of all *.htaccess* files required by the application. It typically looks like this:

```
<Directory /opt/bitnami/apps/mediawiki/htdocs/cache>
  Deny from all
</Directory>
<Directory /opt/bitnami/apps/mediawiki/htdocs/images>
  # Protect against bug 28235
  <IfModule rewrite_module>
    RewriteEngine On
    RewriteCond %{QUERY_STRING} \.[^\\/:*?\x22<>|%]+(#|\?|$) [nocase]
    RewriteRule . - [forbidden]
  </IfModule>
</Directory>
<Directory /opt/bitnami/apps/mediawiki/htdocs/includes>
  Deny from all
</Directory>
<Directory /opt/bitnami/apps/mediawiki/htdocs/languages>
  Deny from all
</Directory>
<Directory /opt/bitnami/apps/mediawiki/htdocs/maintenance>
  Deny from all
</Directory>
<Directory /opt/bitnami/apps/mediawiki/htdocs/maintenance/archives>
  Deny from all
</Directory>
<Directory /opt/bitnami/apps/mediawiki/htdocs/serialized>
  Deny from all
</Directory>
```

How To Create A Password To Protect Access To An Application?

To configure Apache to request a username and password when accessing your application, follow these steps:

- At the console, type the following commands. Remember to replace APPNAME, USERNAME and PASSWORD with your application name, desired username and desired password respectively.

```
$ cd /opt/bitnami
$ apache2/bin/htpasswd -cb apache2/APPNAME_users USERNAME PASSWORD
```

- Edit the */opt/bitnami/apps/APPNAME/conf/httpd-app.conf* file and add the following lines. You also need to comment the *Require all granted* line as shown below:

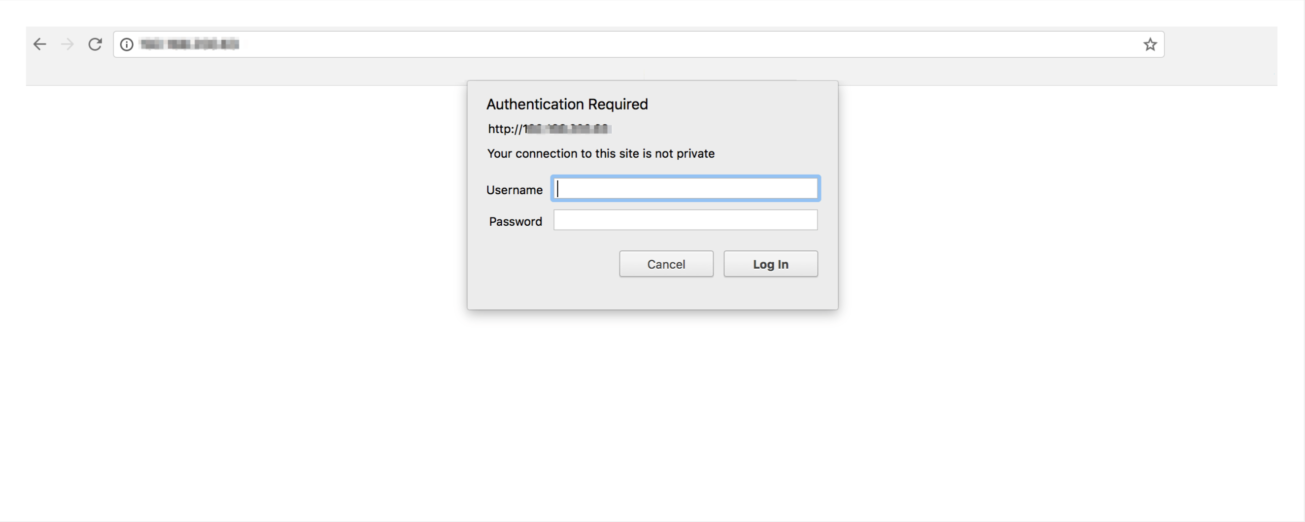
```
<Directory "/opt/bitnami/apps/APPNAME/htdocs">
  ...
    AuthType Basic
    AuthName MyAuthName
    AuthUserFile "/opt/bitnami/apache2/APPNAME_users"
    Require valid-user
  ...

  <IfVersion >= 2.3>
  # Require all granted
  </IfVersion>
  ...
</Directory>
```

- Restart the Apache server:

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

When accessing the application, you will see the following authentication popup window. Enter the username and password that you have defined in the first step:


Authentication Required

http://1[redacted]

Your connection to this site is not private

(/images/img/components/apache/apache-enable-auth-495ed932.png)

To change the password later, run the *htpasswd* utility without the *-c* switch:

```
$ sudo /opt/bitnami/apache2/bin/htpasswd /opt/bitnami/apache2/APPNAME_users USERNAME
```

How To Check If PageSpeed Is Enabled?

To check if PageSpeed is enabled in Apache, refer to this page (/general/components/pagespeed).

How To Access My Application From Only One Domain?

The default Bitnami server configuration allows you to access the server using different methods: using the domain name (eg. *ec2-xx-yy-zz.amazonaws.com* or *xxxx.cloudapp.net* or *xxxx.bitnamiapp.com*), using the IP address directly.

To redirect all these domains to your own domain, add the following configuration into the */opt/bitnami/apache2/conf/bitnami/bitnami.conf* file. Remember to replace *example.com* with your own domain.

```
<VirtualHost _default_:80>
RewriteEngine On
RewriteCond %{HTTP_HOST} !^example.com$
RewriteCond %{HTTP_HOST} !^(localhost|127.0.0.1)
RewriteRule ^(.*)$ http://example.com$1 [R=permanent,L]
...

<VirtualHost _default_:443>
RewriteEngine On
RewriteCond %{HTTP_HOST} !^example.com$
RewriteCond %{HTTP_HOST} !^(localhost|127.0.0.1)
RewriteRule ^(.*)$ https://example.com$1 [R=permanent,L]
...
```

Then, restart the Apache server for the changes to take effect.

Bitnami Documentation

FAQs

How to find application credentials? (/general/faq#find_credentials)
How to connect to the server through SSH? (/general/faq#connect_ssh)
How to upload files to the server with SFTP? (/general/faq#upload_files)
How to open the server ports for remote access? (/general/faq#open_firewall)
How to configure your application to use a third-party SMTP service for outgoing email? (/general/faq#use_external_smtp_title)
How to block a suspicious IP address? (/general/faq#block_suspicious_ip)

Platform Documentation

Google Cloud Platform (/google)
AWS Cloud (/aws)
Oracle Cloud Infrastructure Classic (/oracle)
Microsoft Azure (/azure)
Bitnami Cloud Hosting (/bch)
CenturyLink Cloud (/centurylink)
1&1 Cloud Platform (/1and1)
Huawei Cloud (/huawei)
Open Telekom Cloud (/opentelekomcloud)
Windows / Linux / MacOS (/installer)
Virtual Machines (/virtual-machine)
Containers (/containers)
Kubernetes (/kubernetes)

General Documentation

Bitnami Application Stacks (/general/apps)
Bitnami Infrastructure Stacks (/general/infrastructure)
How-To Guides (/general/how-to)
Bitnami Components (/general/components)
Security Notices (/general/security)

Apps

Applications (https://bitnami.com/stacks)
Add-ons (https://bitnami.com/addons)
Vote! (https://bitnami.com/contest)

What we do

Cloud Hosting (https://bitnami.com/cloud)
Pricing (https://bitnami.com/cloud/pricing)
Enterprise (https://bitnami.com/enterprise)
Cloud Partners (https://bitnami.com/partners/cloud)
Software Partners (https://bitnami.com/partners/software)
Customers (https://bitnami.com/cloud/customers-testimonials)
FAQs (https://bitnami.com/cloud/pricing#faqs)

Who we are

About (https://bitnami.com/about-us)
Contact (https://bitnami.com/contact)
Careers (https://bitnami.com/careers)
What's New? (https://bitnami.com/news)
Press (https://bitnami.com/news/press)
Blog (http://blog.bitnami.com)

Legal (https://bitnami.com/legal)

Support

Documentation (https://bitnami.com/support)
Forums (https://community.bitnami.com)
Helpdesk (http://helpdesk.bitnami.com)
Webinars (https://bitnami.com/webinars)