

httpd is now available in the (<https://store.docker.com/images/3574ef8c-86e4-40ed-a705-3a52d9786bde>) Docker Store, (<https://store.docker.com>) the new place to discover public Docker content. Check it out → (<https://store.docker.com/images/3574ef8c-86e4-40ed-a705-3a52d9786bde>)

(/)

OFFICIAL REPOSITORY

httpd (/ /httpd/) 

Last pushed: 18 days ago

Repo Info (/ /httpd/)

Short Description

The Apache HTTP Server Project

Full Description

Supported tags and respective Dockerfile links

- 2.2.32 , 2.2 (2.2/Dockerfile) (<https://github.com/docker-library/httpd/blob/a5ab6fb434c13a3578a2e7e6ce2cf30d66c625bc/2.2/Dockerfile>)
- 2.2.32-alpine , 2.2-alpine (2.2/alpine/Dockerfile) (<https://github.com/docker-library/httpd/blob/a5ab6fb434c13a3578a2e7e6ce2cf30d66c625bc/2.2/alpine/Dockerfile>)
- 2.4.25 , 2.4 , 2 , latest (2.4/Dockerfile) (<https://github.com/docker-library/httpd/blob/e885c0d63078153dc89fa0212314e590fec7fc93/2.4/Dockerfile>)
- 2.4.25-alpine , 2.4-alpine , 2-alpine , alpine (2.4/alpine/Dockerfile) (<https://github.com/docker-library/httpd/blob/b95f1aba991d613f971fe8c66dc23fb4d8f3e9a7/2.4/alpine/Dockerfile>)

For detailed information about the published artifacts of each of the above supported tags (image metadata, transfer size, etc), please see the repos/httpd directory (<https://github.com/docker-library/repo-info/blob/master/repos/httpd>) in the docker-library/repo-info GitHub repo (<https://github.com/docker-library/repo-info>).

For more information about this image and its history, please see the relevant manifest file (library/httpd) (<https://github.com/docker-library/official-images/blob/master/library/httpd>). This image is updated via pull requests to the docker-library/official-images GitHub repo (<https://github.com/docker-library/official-images>).

images/pulls?q=label%3Alibrary%2Fhttpd).

What is httpd?

The Apache HTTP Server, colloquially called Apache, is a Web server application notable for playing a key role in the initial growth of the World Wide Web. Originally based on the NCSA HTTPd server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache quickly overtook NCSA HTTPd as the dominant HTTP server, and has remained the most popular HTTP server in use since April 1996.

wikipedia.org/wiki/Apache_HTTP_Server
(http://en.wikipedia.org/wiki/Apache_HTTP_Server)



How to use this image.

This image only contains Apache httpd with the defaults from upstream. There is no PHP installed, but it should not be hard to extend. On the other hand, if you just want PHP with Apache httpd see the [PHP image \(https://registry.hub.docker.com/_/php/\)](https://registry.hub.docker.com/_/php/) and look at the `apache` tags. If you want to run a simple HTML server, add a simple Dockerfile to your project where `public-html/` is the directory containing all your HTML.

Create a Dockerfile in your project

```
FROM httpd:2.4
COPY ./public-html/ /usr/local/apache2/htdocs/
```

Then, run the commands to build and run the Docker image:

```
$ docker build -t my-apache2 .
$ docker run -dit --name my-running-app my-apache2
```

Without a Dockerfile

If you don't want to include a `Dockerfile` in your project, it is sufficient to do the following:

```
$ docker run -dit --name my-apache-app -v "$PWD":/usr/local/apac
```

Configuration

To customize the configuration of the httpd server, just `COPY` your custom configuration in as `/usr/local/apache2/conf/httpd.conf`.

```
FROM httpd:2.4
COPY ./my-httpd.conf /usr/local/apache2/conf/httpd.conf
```

SSL/HTTPS

If you want to run your web traffic over SSL, the simplest setup is to `COPY` or `mount (-v)` your `server.crt` and `server.key` into `/usr/local/apache2/conf/` and then customize the `/usr/local/apache2/conf/httpd.conf` by removing the comment from the line with `#Include conf/extra/httpd-ssl.conf`. This config file will use the certificate files previously added and tell the daemon to also listen on port 443. Be sure to also add something like `-p 443:443` to your `docker run` to forward the https port.

The previous steps should work well for development, but we recommend customizing your conf files for production, see <http://httpd.apache.org> (https://httpd.apache.org/docs/2.2/ssl/ssl_faq.html) for more information about SSL setup.

Image Variants

The `httpd` images come in many flavors, each designed for a specific use case.

`httpd:<version>`

This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

`httpd:alpine`

This image is based on the popular [Alpine Linux project \(http://alpinelinux.org\)](http://alpinelinux.org), available in the [alpine official image \(https://hub.docker.com/_/alpine\)](https://hub.docker.com/_/alpine). Alpine Linux is much smaller than most distribution base images (~5MB), and thus leads to much slimmer images in general.

This variant is highly recommended when final image size being as small as possible is desired. The main caveat to note is that it does use [musl libc \(http://www.musl-libc.org\)](http://www.musl-libc.org) instead of [glibc and friends \(http://www.etalabs.net/compare_libcs.html\)](http://www.etalabs.net/compare_libcs.html), so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn't have an issue with this, so this variant is usually a very safe choice. See [this Hacker News comment thread \(https://news.ycombinator.com/item?id=10782897\)](https://news.ycombinator.com/item?id=10782897) for more discussion of the issues that might arise and some pro/con comparisons of using Alpine-based images.

To minimize image size, it's uncommon for additional related tools (such as `git` or `bash`) to be included in Alpine-based images. Using this image as a base, add the things you need in your own Dockerfile (see the [alpine image description \(https://hub.docker.com/_/alpine/\)](https://hub.docker.com/_/alpine/) for examples of how to install packages if you are

unfamiliar).

License

View [license information \(https://www.apache.org/licenses/\)](https://www.apache.org/licenses/) for the software contained in this image.

Supported Docker versions

This image is officially supported on Docker version 17.04.0-ce.

Support for older versions (down to 1.6) is provided on a best-effort basis.

Please see [the Docker installation documentation \(https://docs.docker.com/installation/\)](https://docs.docker.com/installation/) for details on how to upgrade your Docker daemon.

User Feedback

Issues

If you have any problems with or questions about this image, please contact us through a [GitHub issue \(https://github.com/docker-library/httpd/issues\)](https://github.com/docker-library/httpd/issues). If the issue is related to a CVE, please check for a [cve-tracker issue on the official-images repository first \(https://github.com/docker-library/official-images/issues?q=label%3Acve-tracker\)](https://github.com/docker-library/official-images/issues?q=label%3Acve-tracker).

You can also reach many of the official image maintainers via the `#docker-library` IRC channel on [Freenode \(https://freenode.net\)](https://freenode.net).

Contributing

You are invited to contribute new features, fixes, or updates, large or small; we are always thrilled to receive pull requests, and do our best to process them as fast as we can.

Before you start to code, we recommend discussing your plans through a [GitHub issue \(https://github.com/docker-library/httpd/issues\)](https://github.com/docker-library/httpd/issues), especially for more ambitious contributions.

This gives other contributors a chance to point you in the right direction, give you feedback on your design, and help you find out if someone else is working on the same thing.

Documentation

Documentation for this image is stored in the [httpd/ directory \(https://github.com/docker-library/docs/tree/master/httpd\)](https://github.com/docker-library/docs/tree/master/httpd) of the [docker-library/docs GitHub repo \(https://github.com/docker-library/docs\)](https://github.com/docker-library/docs). Be sure to familiarize yourself with the [repository's README.md file \(https://github.com/docker-library/docs/blob/master/README.md\)](https://github.com/docker-library/docs/blob/master/README.md) before attempting a pull request.

Docker Pull Command



```
docker pull httpd
```

Comments (14)

[1](#) [2](#) [»](#)



swyserdev

3 months ago

Do you guys have any idea how I will be able to active http/2 on this image?



owlcyberadmin

3 months ago

Similar to arjanpetersen, I'm getting `"/usr/local/bin/httpd-foreground: 7: exec: httpd: not found"` when attempting to run the httpd image. Has anyone else run into this issue?



icecappacino

5 months ago

Using docker-compose and I get

```
Sun Nov 13 23:43:50.304823 2016] [core:error] [pid 8:tid 140191667947264]
(13)Permission denied: [client 192.168.99.1:61459] AH00132: file permissions deny server
access: /usr/local/apache2/htdocs/img/decibel.png, referer: http://192.168.99.100/?
(http://192.168.99.100/?\(http://192.168.99.100/?\))
```

Why would I get that?



famoustm
8 months ago

I set the port to 5000, however now the person has to come to port 5000?

For instance:

<http://thisismysite.com:5000> (<http://thisismysite.com:5000>)

Of course no one would ever type that, so how can the site be hit without the 5000 port being typed in at the end of the URL?

Thanks for your help!



kinghuang
8 months ago

I submitted [an issue \(https://github.com/docker-library/httpd/issues/27\)](https://github.com/docker-library/httpd/issues/27) and [a pull request \(https://github.com/docker-library/httpd/pull/28\)](https://github.com/docker-library/httpd/pull/28) to fix apxs in the Alpine version of the image over a week ago, but there's been no comments or actions. Is that the correct way to report and fix issues? Or, am I supposed to be doing something else?



arjanpetersen
9 months ago

Hi, I used the dockerfile in the 2.4 directory on the github repository. After building the image - I get

```
arjan@osboxes:~/mydockerbuild$ docker run -it --rm --name my-running-app my-apache2  
docker: Error response from daemon: Container command 'httpd-foreground' not found or  
does not exist..
```

When trying to run the image. Any thoughts. Or should I just stick to commands above?



mgsaballa
9 months ago

I am trying to use ProxyHTMLURLMap and I found out that this image does not have mod_proxy_html.so installed. I found references on how to install it but it seem to require libxml2.so which is also not included on this image.

Does anyone have an idea on how to install this module properly? Thanks.



keithroberts
10 months ago

I tried to install php on this image, but the debian image seems to use a different installation path for apache, so the apache PHP module was installed under:

```
LoadModule php5_module /usr/lib/apache2/modules/libphp5.so
```

There were also separate apache configs installed under /etc/httpd whereas apache was installed under
/usr/local/apache2/

after the following error I gave up trying to get this apache docker image to work:

```
root@piwik2-v1:/usr/local/apache2/conf# apachectl -M
[Wed Jun 15 13:26:40.828711 2016] [:crit] [pid 5741:tid 139654481426304] Apache is
running a threaded MPM, but your PHP Module is not compiled to be threadsafe.
You need to recompile PHP.
AH00013: Pre-configuration failed
```

Maybe it would be better to create the apache docker image by installing the apache distributed with the official debian OS image?

I'm off to try out the official PHP docker image now, to see if that works OK!



nila
10 months ago

ni,

i am trying to set AEM dispatcher using the HTTP docker image.

I am facing issues when loading shared object files (.so) and getting the below error. Please can anyone help?

httpd: Syntax error on line 153 of /usr/local/apache2/conf/httpd.conf: Cannot load modules/mod_dispatcher.so into server: /usr/local/apache2/modules/mod_dispatcher.so: invalid ELF header

No issues with the SO file as it is directly downloaded from AEM site.

I had issues in placing the .so file on the image and container. I managed to download all the modules from container and placed it in local along with the dispatcher.so. On the run command, i have mounted the local volume to the modules dir. All modules are loading fine but the dispatcher module is giving this error.

Any thoughts in placing and loading ".so" file in the container?

i can copy the customised http.conf file successfully.

Thanks & Regards,

Nila

my docker file -



fleporcq

a year ago

To allow apache to start as non-root user :

```
FROM httpd:2.4
```

```
# Create 'me' group whith gid 1000 and 'me' user in this group w
RUN groupadd -f -g 1000 me && useradd -u 1000 -g me me
```

```
# Define apache listen port on a port greater than 1024 to allow
RUN sed -i 's/Listen\ 80/Listen\ 1080/g' /usr/local/apache2/conf,
```

```
# Allow apache to run with 'me' user
RUN chown -R me:me /usr/local/apache2/
```

```
# Start container as me
USER me
```


