openresty/openresty ★

By openresty • Updated 2 days ago

OpenResty Official Docker Images - a dynamic web platform based on NGINX and LuaJIT

Container

docker-openresty - Docker tooling for OpenResty

Supported tags and respective `Dockerfile` links

The following "flavors" are available and built from upstream OpenResty packages:

- `centos` , `centos-rpm` , (*centos/Dockerfile*)
- `stretch` , (*stretch/Dockerfile*)
- `stretch-fat` , (*stretch/Dockerfile.fat*)
- `windows` , (*windows/Dockerfile*)

The following "flavors" are built from source and are intended for more advanced and custom usage, caveat emptor:

- `alpine` , (*alpine/Dockerfile*)
- `alpine-fat` , (*alpine/Dockerfile.fat*)
- `bionic` , (*bionic/Dockerfile*)
- `xenial` , (*xenial/Dockerfile*)

Starting with `1.13.6.1` , releases are tagged with `<openresty-version>-<image-version>-<flavor>` . The latest `image-version` will also be tagged `<openresty-version>-<flavor>` . The HEAD of the master branch is also labeled plainly as `<flavor>` . The builds are managed by Travis-CI and Appveyor (for Windows images).

Starting with `1.15.8.1` , there are also `-nosse42` image flavors for systems which do not support SSE 4.2 (e.g. older systems and embedded systems). They are built with `--build-arg RESTY_CONFIG_OPTIONS_MORE="--with-luajit-xcflags='-mno-sse4.2'"` . It is highly recommended *NOT* to use these if your system supports SSE 4.2 because the `CRC32` instruction dramatically improves large string performance. These are only for built-from-source flavors, e.g. `1.15.8.1-0-bionic-nosse42` , `1.15.8.1-0-alpine-nosse42` , `1.15.8.1-0-alpine-fat-nosse42` .

It is *highly recommended* that you use the upstream-based images for best support. For best stability, pin your images to the full tag, for example `1.15.8.1-0-bionic` .

Table of Contents

Description

`docker-openresty` is Docker tooling for OpenResty (https://www.openresty.org).

Docker is a container management platform.

OpenResty is a full-fledged web application server by bundling the standard nginx core, lots of 3rd-party nginx modules, as well as most of their external dependencies.

Usage

If you are happy with the build defaults, then you can use the openresty image from the Docker Hub. The image tags available there are listed at the top of this README.

```
docker run [options] openresty/openresty:stretch-fat
```

[options] would be things like -p to map ports, -v to map volumes, and -d to daemonize.

`docker-openresty` symlinks `/usr/local/openresty/nginx/logs/access.log` and `error.log` to `/dev/stdout` and `/dev/stderr` respectively, so that Docker logging works correctly. If you change the log paths in your `nginx.conf`, you should symlink those paths as well. This is not possible with the `windows` image.

Nginx Config Files

The Docker tooling installs its own `nginx.conf` file. If you want to directly override it, you can replace it in your own Dockerfile or via volume bind-mounting.

For the Linux images, that `nginx.conf` has the directive `include /etc/nginx/conf.d/*.conf;` so all nginx configurations in that directory will be included. The default virtual host configuration has the original OpenResty configuration and is copied to `/etc/nginx/conf.d/default.conf`.

You can override that `default.conf` directly or volume bind-mount the `/etc/nginx/conf.d` directory to your own set of configurations:

```
docker run -v /my/custom/conf.d:/etc/nginx/conf.d openresty/openresty:alpine
```

When using the `windows` image you can change the main configuration directly:

```
docker run -v C:/my/custom/nginx.conf:C:/openresty/conf/nginx.conf openresty/openresty:windows
```

OPM

Starting at version 1.11.2.2, OpenResty for Linux includes a package manager called `opm`, which can be found at `/usr/local/openresty/bin/opm`.

`opm` is built in all the images except `alpine` and `stretch`.

To use `opm` in the `alpine` image, you must also install the `curl` and `perl` packages; they are not included by default because they double the image size. You may install them like so: `apk add --no-cache curl perl`.

To use `opm` within the `stretch` image, you can either use the `stretch-fat` image or install the `openresty-opm` package in a custom build (which you would need to do to install your own `opm` packages anyway), as shown in this example.

LuaRocks

LuaRocks is included in the `alpine-fat`, `centos`, and `xenial` variants. It is excluded from `alpine` because it generally requires a build system and we want to keep that variant lean.

It is available at `/usr/local/openresty/luajit/bin/luarocks`. Packages can be added in your dependent Dockerfiles like so:

```
RUN /usr/local/openresty/luajit/bin/luarocks install <rock>
```

Tips & Pitfalls

- The `envsubst` utility is included in all images except `alpine` and `windows`; this utility is also included in the Nginx docker image and is used to template environment variables into configuration files.
- **Docker Hub** does not currently support ARM builds, thus an `armhf-xenial` image is not available (See #26). You can build an image yourself using the `RESTY_DEBIAN_BASE` build argument:

  ```
  docker build -f xenial/Dockerfile --build-arg "RESTY_DEBIAN_BASE=armv7/armhf-ubuntu" .
  ```

- By default, OpenResty is built with SSE4.2 optimizations if the build machine supports it. If run on machine without SSE4.2, there will be invalid opcode issues. **Thus all the Docker Hub images require SSE4.2.** You can build a custom image from source explicitly without SSE4.2 support, using build arguments like so:

  ```
  docker build -f xenial/Dockerfile --build-arg "RESTY_CONFIG_OPTIONS_MORE=--with-luajit-xcflags='-mno-sse4.2'" .
  ```

- OpenResty's OpenSSL library version must be compatible with your `opm` and LuaRocks packages' version. At minimum, the numeric portion should be the same (e.g. `1.1.1`). The image label `resty_openssl_version` indicates this value. see Labels.
- The `1.13.6.2-alpine` is built from `OpenSSL 1.0.2r` because of build issues on Alpine. `1.15.8.1-alpine` is built from `OpenSSL 1.1.1c` on `Alpine 3.9`.
- Windows images must be built from the same version as the host system it runs on. See Windows container version compatibility. Our images are currently built from the "Windows Server 2016" series.
- The `SIGQUIT` signal will be sent to nginx to stop this container, to give it an opportunity to stop gracefully (i.e, finish processing active connections). The Docker default is `SIGTERM`, which immediately terminates active connections. Note that if your configuration listens on UNIX domain sockets, this means that you'll need to manually remove the socket file upon shutdown, due to nginx bug #753.

Image Labels

The image builds are labeled with various information, such as the versions of OpenRestyand its dependent libraries. Here's an example of printing the labels using `jq`:

```
$ docker pull openresty/openresty:1.15.8.1-1-alpine
$ docker inspect openresty/openresty:1.15.8.1-1-alpine | jq '.[].Config.Labels'
{
  "maintainer": "Evan Wies <evan@*********.net>",
  "resty_add_package_builddeps": "",
  "resty_add_package_rundeps": "",
  "resty_config_options": "    --with-file-aio     --with-http_addition_module     --with-http_auth_request_module     --with-http_dav_module     --wit
  "resty_config_options_more": "",
  "resty_eval_post_make": "",
  "resty_eval_pre_configure": "",
  "resty_openssl_version": "1.0.2r",
  "resty_pcre_version": "8.42",
  "resty_version": "1.15.8.1"
}
```

Label Name	Description
`maintainer`	Maintainer of the image
`resty_add_package_builddeps`	buildarg `RESTY_ADD_PACKAGE_BUILDDEPS`
`resty_add_package_rundeps`	buildarg `RESTY_ADD_PACKAGE_RUNDEPS`
`resty_config_options`	buildarg `RESTY_CONFIG_OPTIONS`
`resty_config_options_more`	buildarg `RESTY_CONFIG_OPTIONS_MORE`
`resty_deb_flavor`	buildarg `RESTY_DEB_FLAVOR`
`resty_deb_version`	buildarg `RESTY_DEB_VERSION`
`resty_eval_post_make`	buildarg `RESTY_EVAL_POST_MAKE`
`resty_eval_pre_configure`	buildarg `RESTY_EVAL_PRE_CONFIGURE`
`resty_image_base`	Name of the base image to build from, buildarg `RESTY_IMAGE_BASE`
`resty_image_tag`	Tag of the base image to build from, buildarg `RESTY_IMAGE_TAG`
`resty_install_base`	buildarg `RESTY_INSTALL_BASE`
`resty_install_tag`	buildarg `RESTY_INSTALL_TAG`
`resty_luarocks_version`	buildarg `RESTY_LUAROCKS_VERSION`
`resty_openssl_version`	buildarg `RESTY_OPENSSL_VERSION`
`resty_pcre_version`	buildarg `RESTY_PCRE_VERSION`
`resty_rpm_arch`	buildarg `RESTY_RPM_ARCH`
`resty_rpm_flavor`	buildarg `RESTY_RPM_FLAVOR`
`resty_rpm_version`	buildarg `RESTY_RPM_VERSION`
`resty_version`	buildarg `RESTY_VERSION`

Docker CMD

The `-g "daemon off;"` directive is used in the Dockerfile CMD to keep the Nginx daemon running after container creation. If this directive is added to the nginx.conf, then the `docker run` should explicitly invoke `openresty` (or `nginx` for `windows` images):

```
docker run [options] openresty/openresty:xenial openresty
```

Invoke another CMD, for example the `resty` utility, like so:

```
docker run [options] openresty/openresty:xenial resty [script.lua]
```

NOTE The `alpine` images do not include the packages `perl` and `ncurses`, which is needed by the `resty` utility.

Building (from source)

This Docker image can be built and customized by cloning the repo and running `docker build` with the desired Dockerfile:

```
git clone https://github.com/openresty/docker-openresty.git
cd docker-openresty
docker build -t myopenresty -f xenial/Dockerfile .
docker run myopenresty
```

Dockerfiles are provided for the following base systems, selecting the Dockerfile path with `-f`:

- Alpine (`alpine/Dockerfile`)
- Alpine Fat (`alpine-fat/Dockerfile`)
- Ubuntu Xenial (`xenial/Dockerfile`)
- Ubuntu Bionic (`bionic/Dockerfile`)

We used to support more build flavors but have trimmed that down. Older Dockerfiles are archived in the `archive` folder.

The following are the available build-time options. They can be set using the `--build-arg` CLI argument, like so:

```
docker build --build-arg RESTY_J=4 -f xenial/Dockerfile .
```

Key	Default	Description
RESTY_IMAGE_BASE	"ubuntu" / "alpine"	The Debian or Alpine Docker image base to build `FROM`.
RESTY_IMAGE_TAG	{ "xenial", "bionic" } / "3.9"	The Debian or Alpine Docker image tag to build `FROM`.
RESTY_VERSION	1.15.8.1	The version of OpenResty to use.
RESTY_LUAROCKS_VERSION	3.1.3	The version of LuaRocks to use.
RESTY_OPENSSL_VERSION	1.1.0j / 1.1.1c	The version of OpenSSL to use.
RESTY_PCRE_VERSION	8.42	The version of PCRE to use.
RESTY_J	1	Sets the parallelism level (-jN) for the builds.
RESTY_CONFIG_OPTIONS	"--with-file-aio --with-http_addition_module --with-http_auth_request_module --with-http_dav_module --with-http_flv_module --with-http_geoip_module=dynamic --with-http_gunzip_module --with-http_gzip_static_module --with-http_image_filter_module=dynamic --with-http_mp4_module --with-http_perl_module=dynamic --with-http_random_index_module --with-http_realip_module --with-http_secure_link_module --with-http_slice_module --with-http_ssl_module --with-http_stub_status_module --with-http_sub_module --with-http_v2_module --with-http_xslt_module=dynamic --with-ipv6 --with-mail --with-mail_ssl_module --with-md5-asm --with-pcre-jit --with-sha1-asm --with-stream --with-stream_ssl_module --with-threads"	Options to pass to OpenResty's `./configure` script.
RESTY_CONFIG_OPTIONS_MORE	""	More options to pass to OpenResty's `./configure` script.
RESTY_ADD_PACKAGE_BUILDDEPS	""	Additional packages to install with package manager required by build only (removed after installation)
RESTY_ADD_PACKAGE_RUNDEPS	""	Additional packages to install with package manager required at runtime (not removed after installation)
RESTY_EVAL_PRE_CONFIGURE	""	Command(s) to run prior to executing OpenResty's `./configure` script. (this can be used to clone a github repo of an extension you want to add to OpenResty, for example. In that case, dont forget to add the appropriate argument to the RESTY_CONFIG_OPTIONS_MORE argument as described above).
RESTY_EVAL_POST_MAKE	""	Command(s) to run after running make install.

These built-from-source flavors include the following modules by default, but one can easily increase or decrease that with the custom build options above:

- file-aio
- http_addition_module
- http_auth_request_module
- http_dav_module
- http_flv_module
- http_geoip_module=dynamic

- http_gunzip_module
- http_gzip_static_module
- http_image_filter_module=dynamic
- http_mp4_module
- http_random_index_module
- http_realip_module
- http_secure_link_module
- http_slice_module
- http_ssl_module
- http_stub_status_module
- http_sub_module
- http_v2_module
- http_xslt_module=dynamic
- ipv6
- mail
- mail_ssl_module
- md5-asm
- pcre-jit
- sha1-asm
- stream
- stream_ssl_module
- threads Back to TOC

Building (RPM based)

OpenResty now now has RPMs available. The `centos` images use these RPMs rather than building from source.

This Docker image can be built and customized by cloning the repo and running `docker build` with the desired Dockerfile:

- CentOS 7 RPM (`centos/Dockerfile`)

The following are the available build-time options. They can be set using the `--build-arg` CLI argument, like so:

```
docker build --build-arg RESTY_RPM_FLAVOR="-debug" centos
```

Key	Default	Description
RESTY_IMAGE_BASE	"centos"	The Centos Docker image base to build `FROM` .
RESTY_IMAGE_TAG	"7"	The CentOS Docker image tag to build `FROM` .
RESTY_LUAROCKS_VERSION	3.1.2	The version of LuaRocks to use.
RESTY_RPM_FLAVOR	""	The `openresty` package flavor to use. Possibly `"-debug"` or `"-valgrind"` .
RESTY_RPM_VERSION	1.15.8.1-1.el7	The `openresty` package version to install.
RESTY_RPM_ARCH	x86_64	The `openresty` package architecture to install.

Back to TOC

Building (DEB based)

OpenResty now now has Debian Packages (DEBs) available. The `stretch` image use these DEBs rather than building from source.

You can derive your own Docker images from this to install your own packages. See Dockerfile.opm_example and Dockerfile.luarocks_example.

This Docker image can be built and customized by cloning the repo and running `docker build` with the desired Dockerfile:

- Debian Stretch 9 DEB (`stretch/Dockerfile`)

The following are the available build-time options. They can be set using the `--build-arg` CLI argument, like so:

```
docker build --build-arg RESTY_DEB_FLAVOR="-debug" -f stretch/Dockerfile .
```

Key	Default	Description
RESTY_IMAGE_BASE	"debian"	The Debian Docker image base to build `FROM` .
RESTY_IMAGE_TAG	"stretch-slim"	The Debian Docker image tag to build `FROM` .
RESTY_DEB_FLAVOR	""	The `openresty` package flavor to use. Possibly `"-debug"` or `"-valgrind"` .
RESTY_DEB_VERSION	"=1.15.8.1-1~stretch1"	The Debian package version to use, with `=` prepended.

Back to TOC

Building (Windows based)

This Docker image can be built and customized by cloning the repo and running `docker build` with the desired Dockerfile:

- Windows (`windows/Dockerfile`)

The following are the available build-time options. They can be set using the `--build-arg` CLI argument, like so:

```
docker build --build-arg RESTY_VERSION="1.13.6.2" -f windows/Dockerfile .
```

Key	Default	Description
RESTY_INSTALL_BASE	"microsoft/windowsservercore"	The Windows Server Docker image name to download and install OpenResty with.
RESTY_INSTALL_TAG	"ltsc2016"	The Windows Server Docker image name to download and install OpenResty with.
RESTY_IMAGE_BASE	"microsoft/nanoserver"	The Windows Server Docker image name to build `FROM` for final image.
RESTY_IMAGE_TAG	"sac2016"	The Windows Server Docker image tag to build `FROM` for final image.
RESTY_VERSION	1.15.8.1	The version of OpenResty to use.

Back to TOC

Feedback & Bug Reports

You're very welcome to report bugs and give feedback as GitHub Issues:

https://github.com/openresty/docker-openresty/issues

Back to TOC

Changelog & Authors

- CHANGELOG
- AUTHORS

Back to TOC

Copyright & License

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Back to TOC /evan@neomantra.net</evan@*****.net>

Docker Pull Command

Copied!

```
docker pull openresty/openresty
```

Owner

openresty

Source Repository

Github
openresty/docker-openresty