

---

# Dockerfile Documentation

**Florian Tatzel**

**Dec 16, 2018**



---

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What are the Dockerfile for? . . . . .	3
<b>2</b>	<b>Docker images</b>	<b>5</b>
2.1	webdevops/ansible . . . . .	5
2.2	webdevops/apache . . . . .	6
2.3	webdevops/apache-dev . . . . .	7
2.4	webdevops/base . . . . .	9
2.5	webdevops/base-app . . . . .	12
2.6	webdevops/bootstrap . . . . .	13
2.7	webdevops/certbot . . . . .	14
2.8	webdevops/hhvm . . . . .	14
2.9	webdevops/hhvm-apache . . . . .	15
2.10	webdevops/hhvm-nginx . . . . .	17
2.11	webdevops/liquibase . . . . .	19
2.12	webdevops/liquidsoap . . . . .	20
2.13	webdevops/mail-sandbox . . . . .	22
2.14	webdevops/nginx . . . . .	24
2.15	webdevops/nginx-dev . . . . .	26
2.16	webdevops/php . . . . .	28
2.17	webdevops/php-apache . . . . .	31
2.18	webdevops/php-apache-dev . . . . .	35
2.19	webdevops/php-dev . . . . .	40
2.20	webdevops/php-nginx . . . . .	45
2.21	webdevops/php-nginx-dev . . . . .	49
2.22	webdevops/postfix . . . . .	54
2.23	webdevops/samson-deployment . . . . .	55
2.24	webdevops/sphinx . . . . .	56
2.25	webdevops/ssh . . . . .	57
2.26	webdevops/storage . . . . .	57
2.27	webdevops/typo3-solr . . . . .	57
2.28	webdevops/varnish . . . . .	58
2.29	webdevops/vsftp . . . . .	58
<b>3</b>	<b>Commands (bin/console)</b>	<b>59</b>
3.1	Requirements . . . . .	59
3.2	Install dependencies . . . . .	59

3.3	Configuration . . . . .	59
3.4	bin/console tasks . . . . .	59
<b>4</b>	<b>Customization</b>	<b>63</b>
4.1	Docker image tools . . . . .	63
4.2	Provisioning . . . . .	64
4.3	Supervisor Daemon (Services) . . . . .	65
<b>5</b>	<b>Known issues</b>	<b>67</b>
5.1	General images . . . . .	67
5.2	PHP images . . . . .	67
<b>6</b>	<b>FAQ</b>	<b>69</b>

Welcome to the documentation pages for the WebDevOps Dockerfiles.

Here you will find informations regarding out Dockerfiles and how to use them.

This documentation is work in progress and we appreciate every help we can get, so feel free to contribute.



### 1.1 What are the Dockerfile for?

The Dockerfile repository is responsible for generating our Docker images on <https://hub.docker.com/u/webdevops/>.

We're generating Docker images for PHP, Apache, Nginx, Varnish or Sphinx.

All Docker images are generated by our Build server at <https://build.webdevops.io/>.





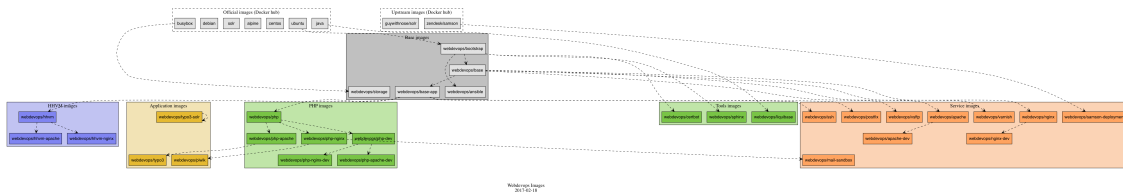


Fig. 1: Docker image hierarchy

## 2.1 webdevops/ansible

All webdevops/ansible images are based on webvdevops/bootstrap without any changes.

### 2.1.1 Docker image tags

Tag	Distribution name
alpine	Alpine (rolling release)
alpine-3	<i>deprecated</i>
ubuntu-12.04	precise
ubuntu-14.04	trusty (LTS)
ubuntu-15.04	vivid
ubuntu-15.10	wily
ubuntu-16.04	xenial (LTS)
debian-7	wheezy
debian-8	jessie
debian-9	stretch
centos-7	

## 2.2 webdevops/apache

These image extends `webdevops/base` with a apache daemon which is running on port 80 and 443

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.2.1 Docker image tags

Tag	Distribution name
<code>alpine</code>	Alpine (rolling release)
<code>alpine-3</code>	<i>deprecated</i>
<code>ubuntu-12.04</code>	precise
<code>ubuntu-14.04</code>	trusty (LTS)
<code>ubuntu-15.04</code>	vivid
<code>ubuntu-15.10</code>	wily
<code>ubuntu-16.04</code>	xenial (LTS)
<code>debian-7</code>	wheezy
<code>debian-8</code>	jessie
<code>debian-9</code>	stretch
<code>centos-7</code>	

### 2.2.2 Environment variables

#### Web environment variables

Environment variable	Description	Default
<code>WEB_DOCUMENT_ROOT</code>	Document root for webserver	<code>/app</code>
<code>WEB_DOCUMENT_INDEX</code>	Index document	<code>index.php</code>
<code>WEB_ALIAS_DOMAIN</code>	Domain aliases	<code>*.vm</code>
<code>WEB_PHP_SOCKET</code>	PHP-FPM socket address	<code>127.0.0.1:9000</code> (for <code>php-*</code> images)
<code>SERVICE_PHPFPM_OPTS</code>	PHP-FPM command arguments	<i>empty</i> (when <code>php fpm</code> is used)
<code>SERVICE_APACHE_OPTS</code>	Apache command arguments	<i>empty</i> (when <code>apache</code> is used)
<code>SERVICE_NGINX_OPTS</code>	Nginx command arguments	<i>empty</i> (when <code>nginx</code> is used)

### 2.2.3 Customization

#### Apache customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/httpd/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/httpd/vhost.common.d` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.2.4 Docker image layout

### Apache layout

File/Directory	Description
/opt/docker/etc/httpd/conf.d	Main global configuration directory (automatically included files)
/opt/docker/etc/httpd/conf.d/10-php.conf	PHP cgi configuration
/opt/docker/etc/httpd/conf.d/10-error-document.conf	Error document configuration
/opt/docker/etc/httpd/conf.d/10-log.conf	Log configuration
/opt/docker/etc/httpd/conf.d/10-server.conf	Basic server configuration
/opt/docker/etc/httpd/ssl	SSL configuration directory for certifications and keys
/opt/docker/etc/httpd/ssl/server.crt	Example SSL certification (*.vm)
/opt/docker/etc/httpd/ssl/server.csr	Example SSL certification request (*.vm)
/opt/docker/etc/httpd/ssl/server.key	Example SSL key (*.vm)
/opt/docker/etc/httpd/vhost.common.d	Vhost configuration directory (automatically included files)
/opt/docker/etc/httpd/vhost.common.d/01-boilerplate.conf	Placeholder configuration file (prevent include errors for Apache 2.2)
/opt/docker/etc/httpd/global.conf	Global httpd configuration
/opt/docker/etc/httpd/main.conf	Main httpd configuration
/opt/docker/etc/httpd/php.conf	Deprecated PHP configuration
/opt/docker/etc/httpd/vhost.common.conf	Deprecated vhost common include
/opt/docker/etc/httpd/vhost.conf	Vhost configuration
/opt/docker/etc/httpd/vhost.ssl.conf	Vhost SSL configuration
/opt/docker/etc/supervisor.d/httpd.conf	Supervisord configuration file for Apache HTTPD

## 2.3 webdevops/apache-dev

These image extends `webdevops/base` with a apache daemon which is running on port 80 and 443

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

**Attention:** Webserver is running development mode, assets and resource files will be send without caching to browser!

### 2.3.1 Docker image tags

Tag	Distribution name
alpine	Alpine (rolling release)
alpine-3	<i>deprecated</i>
ubuntu-12.04	precise
ubuntu-14.04	trusty (LTS)
ubuntu-15.04	vivid
ubuntu-15.10	wily
ubuntu-16.04	xenial (LTS)
debian-7	wheezy
debian-8	jessie
debian-9	stretch
centos-7	

### 2.3.2 Environment variables

#### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

#### Web development environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
WEB_PHP_TIMEOUT	Webserver PHP request timeout	600
WEB_NO_CACHE_PATTERNS	RegEx of files which should be delivered by webserver as non cacheable to browser	\.(css js gif png jpg svg json xml)\$

### 2.3.3 Customization

#### Apache customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/httpd/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/httpd/vhost.common.d` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.3.4 Docker image layout

### Apache layout

File/Directory	Description
/opt/docker/etc/httpd/conf.d	Main global configuration directory (automatically included files)
/opt/docker/etc/httpd/conf.d/10-php.conf	PHP cgi configuration
/opt/docker/etc/httpd/conf.d/10-error-document.conf	Error document configuration
/opt/docker/etc/httpd/conf.d/10-log.conf	Log configuration
/opt/docker/etc/httpd/conf.d/10-server.conf	Basic server configuration
/opt/docker/etc/httpd/ssl	SSL configuration directory for certifications and keys
/opt/docker/etc/httpd/ssl/server.crt	Example SSL certification (*.vm)
/opt/docker/etc/httpd/ssl/server.csr	Example SSL certification request (*.vm)
/opt/docker/etc/httpd/ssl/server.key	Example SSL key (*.vm)
/opt/docker/etc/httpd/vhost.common.d	Vhost configuration directory (automatically included files)
/opt/docker/etc/httpd/vhost.common.d/01-boilerplate.conf	Placeholder configuration file (prevent include errors for Apache 2.2)
/opt/docker/etc/httpd/global.conf	Global httpd configuration
/opt/docker/etc/httpd/main.conf	Main httpd configuration
/opt/docker/etc/httpd/php.conf	Deprecated PHP configuration
/opt/docker/etc/httpd/vhost.common.conf	Deprecated vhost common include
/opt/docker/etc/httpd/vhost.conf	Vhost configuration
/opt/docker/etc/httpd/vhost.ssl.conf	Vhost SSL configuration
/opt/docker/etc/supervisor.d/httpd.conf	Supervisord configuration file for Apache HTTPD

## 2.4 webdevops/base

Our application base container contains some general tools, the provisioning system (Ansible), a preconfigured modular supervisord and a modular *entrypoint* script.

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

## 2.4.1 Docker image tags

Tag	Distribution name
alpine	Alpine (rolling release)
alpine-3	<i>deprecated</i>
ubuntu-12.04	precise
ubuntu-14.04	trusty (LTS)
ubuntu-15.04	vivid
ubuntu-15.10	wily
ubuntu-16.04	xenial (LTS)
debian-7	wheezy
debian-8	jessie
debian-9	stretch
centos-7	

## 2.4.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

## 2.4.3 Entrypoint

The entrypoint script is located in `/opt/docker/bin/entrypoint.sh` and will also start the entrypoint provisioning before running the requested CMD.

Based on CMD the entrypoint script is trying to find the appropriate worker script located in `/opt/docker/bin/entrypoint.d` and executes it. It must matches the CMD and if there is no appropriate worker script the entrypoint falls back to `/opt/docker/bin/entrypoint.d/default.sh` which just executes the specified CMD.

This approach allows a modular entrypoint and also allows to directly jump into a container (eg. with `docker run -ti webdevops/apache bash`) without uploading or modifying the entrypoint.

Example for starting supervisord (executed by `entrypoint.d/supervisord.sh`):

```
ENTRYPOINT ["/opt/docker/bin/entrypoint.sh"]
CMD ["supervisord"]
```

File/Directory	Description
/opt/docker/bin/entrypoint.sh	Entrypoint script for ENTRYPOINT instruction in Dockerfile
/opt/docker/bin/entrypoint.d/default.sh	Default worker script, will execute the defined cmd
/opt/docker/bin/entrypoint.d/cli.sh	Starts predefined CLI_SCRIPT (environment variable)
/opt/docker/bin/entrypoint.d/noop.sh	Starts a noop endless loop at startup
/opt/docker/bin/entrypoint.d/root.sh	Starts a root shell (deprecated)
/opt/docker/bin/entrypoint.d/supervisord.sh	Starts supervisord daemon at startup (CMD ["supervisord"])

## 2.4.4 Supervisord

Supervisord is a lightweight daemon which starts and monitor other programs. We're using it for running more than one task in a docker container (eg. PHP-FPM and Apache/Nginx).

File/Directory	Description
/opt/docker/etc/supervisor.conf	Main supervisord configuration file
/opt/docker/etc/supervisor.d/*.conf	Modular service configuration files for supervisord (will be included automatically)
/opt/docker/bin/serve.d/*.sh	Service scripts if services needs more than just a single command line for startup

## 2.4.5 Provisioning

With Ansible the provisioning tasks can be easily done inside the Docker image (eg. for configurations and deployments). There is also a small provision script for registering and running Ansible roles.

For even simpler tasks there is also a possibility to upload small shell scripts which will be executed at the specific tags.

Register a new role (eg. with tag build): `/opt/docker/bin/provision add --tag build rolename`

Run all registered roles and scripts (in Dockerfile): `/opt/docker/bin/bootstrap.sh`

Tag	Description
bootstrap	Run on Docker image creation (only run once)
build	Run on Docker image build
onbuild	Run on Docker image ONBUILD
entrypoint	Run on Docker image entrypoint execution (only here Environment variables are available)

File/Directory	Description
/opt/docker/bin/provision	Provision script
/opt/docker/bin/bootstrap.sh	Wrapper for running registred provisions (just run it as last script in Dockerfile)
/opt/docker/provision/roles	Directory for Ansible roles
/opt/docker/provision/ bootstrap.d/*.sh	Directory for provisioning shell scripts (tag: bootstrap)
/opt/docker/provision/build.d/ *.sh	Directory for provisioning shell scripts (tag: build)
/opt/docker/provision/onbuild. d/*.sh	Directory for provisioning shell scripts (tag: onbuild)
/opt/docker/provision/ entrypoint.d/*.sh	Directory for provisioning shell scripts (tag: entrypoint)

## 2.5 webdevops/base-app

The base-app image extends the base image with additional tools and all locales.

Packages:

- OpenSSH server (disabled by default) and client
- MySQL client
- sqlite
- dnsmasq (disabled by default)
- postfix (disabled by default)
- sudo
- zip, unzip, bzip2
- wget, curl
- net-tools
- moreutils
- dns utils
- rsync
- git
- nano, vim

Because some applications are using locales for translations (eg. date formatting) all locales are generated inside this image.

For an example docker service

**Attention:** Alpine doesn't provide any locales so you have to find another method for using locales!

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.



## 2.5.1 Docker image tags

Tag	Distribution name
alpine	Alpine (rolling release)
alpine-3	<i>deprecated</i>
ubuntu-12.04	precise
ubuntu-14.04	trusty (LTS)
ubuntu-15.04	vivid
ubuntu-15.10	wily
ubuntu-16.04	xenial (LTS)
debian-7	wheezy
debian-8	jessie
debian-9	stretch
centos-7	

## 2.5.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

## 2.6 webdevops/bootstrap

Bootstrap images contains our baselayout (some basic scripts for secure and small package installations and handling) and a basic toolset. It will also install Ansible into the container for future provisioning of the container.

## 2.6.1 Docker image tags

Tag	Distribution name
alpine	Alpine (rolling release)
alpine-3	<i>deprecated</i>
ubuntu-12.04	precise
ubuntu-14.04	trusty (LTS)
ubuntu-15.04	vivid
ubuntu-15.10	wily
ubuntu-16.04	xenial (LTS)
debian-7	wheezy
debian-8	jessie
debian-9	stretch
centos-7	

## 2.6.2 Baselayout scripts

File/Directory	Description
/usr/local/bin/apk-install	<b>Alpine:</b> Updates package cache, install packages and clears package cache
/usr/local/bin/apk-upgrade	<b>Alpine:</b> Run package upgrade
/usr/local/bin/apt-install	<b>Debian family:</b> Updates package cache, install packages and clears package cache
/usr/local/bin/apt-upgrade	<b>Debian family:</b> Run package upgrade
/usr/local/bin/yum-install	<b>RedHat family:</b> Updates package cache, install packages and clears package cache
/usr/local/bin/yum-upgrade	<b>RedHat family:</b> Run package upgrade
/usr/local/bin/generate-locales	Generate locales
/usr/local/bin/rpl	Script which can replace text in files
/usr/local/bin/service	Supervisord service wrapper script (like <code>service</code> in Debian)

## 2.7 webdevops/certbot

The certbot images are based on webdevops/bootstrap with let's encrypt certbot toolbox.

### 2.7.1 Docker image tags

Tag	Distribution name
latest	Alpine 3

## 2.8 webdevops/hhvm

The hhvm images are based on webdevops/base-app with HHVM cli and HHVM daemon. HHVM daemon is running on port 9000.

## Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.8.1 Docker image tags

Tag	Distribution name
<code>ubuntu-14.04</code>	trusty (LTS)
<code>ubuntu-16.04</code>	xenial (LTS)

### 2.8.2 Environment variables

#### Base environment variables

Environment variable	Description	Default
<code>LOG_STDOUT</code>	Destination of daemon output	<i>empty</i> (stdout)
<code>LOG_STDERR</code>	Destination of daemon errors	<i>empty</i> (stdout)
<code>SERVICE_CRON_OPTS</code>	cron daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_OPTS</code>	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_USER</code>	dnsmasq effective user	<code>root</code>
<code>SERVICE_POSTFIX_OPTS</code>	postfix daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SSH_OPTS</code>	ssh daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_OPTS</code>	supervisor daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_USER</code>	supervisor effective user	<code>root</code>
<code>SERVICE_SYSLOG_OPTS</code>	syslog daemon arguments	<i>empty</i> (when syslog is used)

#### Base environment variables

Environment variable	Description	Default
<code>POSTFIX_MYNETWORKS</code>	Postfix mynetworks address	<i>empty</i>
<code>POSTFIX_RELAYHOST</code>	Postfix upstream relay server	<i>empty</i>

### 2.8.3 Docker image layout

#### HHVM layout

File/Directory	Description
<code>/opt/docker/etc/supervisor.d/hhvm.conf</code>	Supervisord configuration file for HHVM

## 2.9 webdevops/hhvm-apache

This image extends `webdevops/hhvm` with a apache daemon which is running on port 80 and 443

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

## 2.9.1 Docker image tags

Tag	Distribution name
ubuntu-14.04	trusty (LTS)
ubuntu-16.04	xenial (LTS)

## 2.9.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

### Web environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
SERVICE_PHPFPM_OPTS	PHP-FPM command arguments	<i>empty</i> (when php fpm is used)
SERVICE_APACHE_OPTS	Apache command arguments	<i>empty</i> (when apache is used)
SERVICE_NGINX_OPTS	Nginx command arguments	<i>empty</i> (when nginx is used)

## 2.9.3 Customization

### Apache customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/httpd/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/httpd/vhost.common.d` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.9.4 Docker image layout

### Apache layout

File/Directory	Description
<code>/opt/docker/etc/httpd/conf.d</code>	Main global configuration directory (automatically included files)
<code>/opt/docker/etc/httpd/conf.d/10-php.conf</code>	PHP cgi configuration
<code>/opt/docker/etc/httpd/conf.d/10-error-document.conf</code>	Error document configuration
<code>/opt/docker/etc/httpd/conf.d/10-log.conf</code>	Log configuration
<code>/opt/docker/etc/httpd/conf.d/10-server.conf</code>	Basic server configuration
<code>/opt/docker/etc/httpd/ssl</code>	SSL configuration directory for certifications and keys
<code>/opt/docker/etc/httpd/ssl/server.crt</code>	Example SSL certification (*.vm)
<code>/opt/docker/etc/httpd/ssl/server.csr</code>	Example SSL certification request (*.vm)
<code>/opt/docker/etc/httpd/ssl/server.key</code>	Example SSL key (*.vm)
<code>/opt/docker/etc/httpd/vhost.common.d</code>	Vhost configuration directory (automatically included files)
<code>/opt/docker/etc/httpd/vhost.common.d/01-boilerplate.conf</code>	Placeholder configuration file (prevent include errors for Apache 2.2)
<code>/opt/docker/etc/httpd/global.conf</code>	Global httpd configuration
<code>/opt/docker/etc/httpd/main.conf</code>	Main httpd configuration
<code>/opt/docker/etc/httpd/php.conf</code>	Deprecated PHP configuration
<code>/opt/docker/etc/httpd/vhost.common.conf</code>	Deprecated vhost common include
<code>/opt/docker/etc/httpd/vhost.conf</code>	Vhost configuration
<code>/opt/docker/etc/httpd/vhost.ssl.conf</code>	Vhost SSL configuration
<code>/opt/docker/etc/supervisor.d/httpd.conf</code>	Supervisord configuration file for Apache HTTPD

### HHVM layout

File/Directory	Description
<code>/opt/docker/etc/supervisor.d/hhvm.conf</code>	Supervisord configuration file for HHVM

## 2.10 webdevops/hhvm-nginx

This image extends `webdevops/hhvm` with a `nginx` daemon which is running on port 80 and 443

## Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.10.1 Docker image tags

Tag	Distribution name
<code>ubuntu-14.04</code>	trusty (LTS)
<code>ubuntu-16.04</code>	xenial (LTS)

### 2.10.2 Environment variables

#### Base environment variables

Environment variable	Description	Default
<code>LOG_STDOUT</code>	Destination of daemon output	<i>empty</i> (stdout)
<code>LOG_STDERR</code>	Destination of daemon errors	<i>empty</i> (stdout)
<code>SERVICE_CRON_OPTS</code>	cron daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_OPTS</code>	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_USER</code>	dnsmasq effective user	<code>root</code>
<code>SERVICE_POSTFIX_OPTS</code>	postfix daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SSH_OPTS</code>	ssh daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_OPTS</code>	supervisor daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_USER</code>	supervisor effective user	<code>root</code>
<code>SERVICE_SYSLOG_OPTS</code>	syslog daemon arguments	<i>empty</i> (when syslog is used)

#### Base environment variables

Environment variable	Description	Default
<code>POSTFIX_MYNETWORKS</code>	Postfix mynetworks address	<i>empty</i>
<code>POSTFIX_RELAYHOST</code>	Postfix upstream relay server	<i>empty</i>

#### Web environment variables

Environment variable	Description	Default
<code>WEB_DOCUMENT_ROOT</code>	Document root for webserver	<code>/app</code>
<code>WEB_DOCUMENT_INDEX</code>	Index document	<code>index.php</code>
<code>WEB_ALIAS_DOMAIN</code>	Domain aliases	<code>*.vm</code>
<code>WEB_PHP_SOCKET</code>	PHP-FPM socket address	<code>127.0.0.1:9000</code> (for php-* images)
<code>SERVICE_PHPFPM_OPTS</code>	PHP-FPM command arguments	<i>empty</i> (when php fpm is used)
<code>SERVICE_APACHE_OPTS</code>	Apache command arguments	<i>empty</i> (when apache is used)
<code>SERVICE_NGINX_OPTS</code>	Nginx command arguments	<i>empty</i> (when nginx is used)

## 2.10.3 Customization

### Ngix customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/nginx/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/nginx/vhost.common.conf` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.10.4 Docker image layout

### Ngix layout

File/Directory	Description
<code>/opt/docker/etc/nginx/conf.d</code>	Main global configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/conf.d/10-php.conf</code>	PHP cgi configuration
<code>/opt/docker/etc/nginx/ssl</code>	SSL configuration directory for certifications and keys
<code>/opt/docker/etc/nginx//ssl/server.crt</code>	Example SSL certification (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.csr</code>	Example SSL certification request (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.key</code>	Example SSL key (*.vm)
<code>/opt/docker/etc/nginx/vhost.common.d</code>	Vhost configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/vhost.common.d/10-location-root.conf</code>	Redirect requests to DOCUMENT_INDEX
<code>/opt/docker/etc/nginx/vhost.common.d/10-php.conf</code>	PHP cgi configuration for vhost
<code>/opt/docker/etc/nginx/global.conf</code>	Global nginx configuration
<code>/opt/docker/etc/nginx/main.conf</code>	Main Nginx configuration
<code>/opt/docker/etc/nginx/php.conf</code>	Deprecated PHP configuration
<code>/opt/docker/etc/nginx/vhost.common.conf</code>	Deprecated vhost common include
<code>/opt/docker/etc/nginx/vhost.conf</code>	Vhost configuration
<code>/opt/docker/etc/nginx/vhost.ssl.conf</code>	Vhost SSL configuration
<code>/opt/docker/etc/supervisor.d/nginx.conf</code>	Supervisord configuration file for Nginx

### HHVM layout

File/Directory	Description
<code>/opt/docker/etc/supervisor.d/hhvm.conf</code>	Supervisord configuration file for HHVM

## 2.11 webdevops/liquibase

The liquibase images are based on `java` with liquibase and mysql driver

### 2.11.1 Docker image tags

Tag	Distribution name
latest	Based on official java with mysql support
mysql	Based on official java with mysql support
postgres	Based on official java with postgres support

### 2.11.2 Environment variables

Environment variable	Description	Default
LIQUIBASE_VERSION	Installed Liquibase version	<i>not changeable</i>
LIQUIBASE_DRIVER	Database driver	<i>based on docker tag</i>
LIQUIBASE_CLASSPATH	Java class path	<i>based on docker tag</i>
LIQUIBASE_URL	DB url	<i>empty</i> (eg. jdbc:mysql://host/app)
LIQUIBASE_USERNAME	DB username	<i>empty</i>
LIQUIBASE_PASSWORD	DB password	<i>empty</i>
LIQUIBASE_CHANGELOG	Changelog file	/liquibase/changelog.xml
LIQUIBASE_CONTEXTS	Server contexts	<i>empty</i>
LIQUIBASE_OPTS	Additional options	<i>empty</i>

### 2.11.3 Usage

**MYSQL:** Expecting the changelog.xml is inside the current directory the update process can be started with:

```
docker run --rm -v $(pwd):/liquibase/ -e "LIQUIBASE_URL=jdbc:mysql://host/app" -e "LIQUIBASE_USERNAME=root" -e "LIQUIBASE_PASSWORD=root" webdevops/liquibase:mysql update
```

**POSTGRES:** Expecting the changelog.xml is inside the current directory the update process can be started with:

```
docker run --rm -v $(pwd):/liquibase/ -e "LIQUIBASE_URL=jdbc:postgresql://host:5432/sampledb" -e "LIQUIBASE_USERNAME=root" -e "LIQUIBASE_PASSWORD=root" webdevops/liquibase:postgres update
```

## 2.12 webdevops/liquidsoap

The liquidsoap images are based on webdevops/base with liquidsoap multimedia streaming server with most plugins.

### Uses Supervisor

This image is using supervisor and runs the daemon under user application (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.12.1 Docker image tags

Tag	Distribution name
latest	Based on <i>webdevops/base:latest</i> (Ubuntu)



## 2.12.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

Liquisoap environment variables

Environment variable	Description	Default
LIQUIDSOAP_USER	Daemon user ID	liquisoap
LIQUIDSOAP_TELNET	Open telnet (port 1234)	1 (enabled)
LIQUIDSOAP_SCRIPT	Configuration script for liquisoap	/opt/docker/etc/liquisoap/default.liq
LIQUIDSOAP_TEMPLATE	Apply template to config script	1/opt/docker/etc/liquisoap/default.liq
LIQUIDSOAP_STREAM	Input stream (eg. icecast)	http://icecast:8000/live
LIQUIDSOAP_PLAYLIST_DEFAULT	Default stream when no other stream is active	audio_to_stereo(single('/opt/docker/etc/liquisoap/default.mp3'))
LIQUIDSOAP_PLAYLIST_DAY	Daytime stream/playlist	playlist('/opt/docker/etc/liquisoap/playlist-day.pls')
LIQUIDSOAP_PLAYLIST_DAY_TIME	Time range for day playlist	TIME RANGE
LIQUIDSOAP_PLAYLIST_NIGHT	Nighttime stream/playlist	playlist('/opt/docker/etc/liquisoap/playlist-night.pls')
LIQUIDSOAP_PLAYLIST_NIGHT_TIME	Time range for night playlist	TIME RANGE
LIQUIDSOAP_OUTPUT	Output stream (eg. icecast)	output.icecast(%mp3(bitrate=128),host='localhost',port=8000,password='secretpassword',mount='liquisoap-128',name=META_name,genre=META_genre,url=META_url,description=META_desc,ALL_input)
LIQUIDSOAP_OUTPUT_1 ... LIQUIDSOAP_OUTPUT_20	More output stream lines	empty
LIQUIDSOAP_METADATA	Station name	Liquisoap Docker
LIQUIDSOAP_METADATA_GENRE	Station genre	empty
LIQUIDSOAP_METADATA_URL	Station url	empty
LIQUIDSOAP_METADATA_DESCRIPTION	Station description	empty

2.13 webdevops/mail-sandbox

These image extends webdevops/base with a postfix daemon which is running on port 25 and dovecot on IMAP.

This images catches all emails sent to it and stores them locally. These mails are available via IMAP and web (roundcube)

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

## 2.13.1 Docker image tags

Tag	Distribution name
latest	Ubuntu 18.04 xenial (LTS)

## 2.13.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

### Mail sandbox environment variables

Environment variable	Description	Default
MAILBOX_USERNAME	IMAP user	dev
MAILBOX_PASSWORD	IMAP user password	dev

### 2.13.3 Docker image layout

## 2.14 webdevops/nginx

This image extends `webdevops/base` with a `nginx` daemon which is running on port 80 and 443

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.14.1 Docker image tags

Tag	Distribution name
<code>alpine</code>	Alpine (rolling release)
<code>alpine-3</code>	<i>deprecated</i>
<code>ubuntu-12.04</code>	precise
<code>ubuntu-14.04</code>	trusty (LTS)
<code>ubuntu-15.04</code>	vivid
<code>ubuntu-15.10</code>	wily
<code>ubuntu-16.04</code>	xenial (LTS)
<code>debian-7</code>	wheezy
<code>debian-8</code>	jessie
<code>debian-9</code>	stretch
<code>centos-7</code>	

### 2.14.2 Environment variables

#### Base environment variables

Environment variable	Description	Default
<code>LOG_STDOUT</code>	Destination of daemon output	<i>empty</i> (stdout)
<code>LOG_STDERR</code>	Destination of daemon errors	<i>empty</i> (stdout)
<code>SERVICE_CRON_OPTS</code>	cron daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_OPTS</code>	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_USER</code>	dnsmasq effective user	root
<code>SERVICE_POSTFIX_OPTS</code>	postfix daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SSH_OPTS</code>	ssh daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_OPTS</code>	supervisor daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_USER</code>	supervisor effective user	root
<code>SERVICE_SYSLOG_OPTS</code>	syslog daemon arguments	<i>empty</i> (when syslog is used)

## Web environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
SERVICE_PHPFPM_OPTS	PHP-FPM command arguments	<i>empty</i> (when php fpm is used)
SERVICE_APACHE_OPTS	Apache command arguments	<i>empty</i> (when apache is used)
SERVICE_NGINX_OPTS	Nginx command arguments	<i>empty</i> (when nginx is used)

## 2.14.3 Customization

### Nginx customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/nginx/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/nginx/vhost.common.conf` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.14.4 Docker image layout

### Nginx layout

File/Directory	Description
<code>/opt/docker/etc/nginx/conf.d</code>	Main global configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/conf.d/10-php.conf</code>	PHP cgi configuration
<code>/opt/docker/etc/nginx/ssl</code>	SSL configuration directory for certifications and keys
<code>/opt/docker/etc/nginx//ssl/server.crt</code>	Example SSL certification (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.csr</code>	Example SSL certification request (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.key</code>	Example SSL key (*.vm)
<code>/opt/docker/etc/nginx/vhost.common.d</code>	Vhost configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/vhost.common.d/10-location-root.conf</code>	Redirect requests to DOCUMENT_INDEX
<code>/opt/docker/etc/nginx/vhost.common.d/10-php.conf</code>	PHP cgi configuration for vhost
<code>/opt/docker/etc/nginx/global.conf</code>	Global nginx configuration
<code>/opt/docker/etc/nginx/main.conf</code>	Main Nginx configuration
<code>/opt/docker/etc/nginx/php.conf</code>	Deprecated PHP configuration
<code>/opt/docker/etc/nginx/vhost.common.conf</code>	Deprecated vhost common include
<code>/opt/docker/etc/nginx/vhost.conf</code>	Vhost configuration
<code>/opt/docker/etc/nginx/vhost.ssl.conf</code>	Vhost SSL configuration
<code>/opt/docker/etc/supervisor.d/nginx.conf</code>	Supervisord configuration file for Nginx

## 2.15 webdevops/nginx-dev

This image extends `webdevops/base` with a `nginx` daemon which is running on port 80 and 443

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

**Attention:** Webserver is running development mode, assets and resource files will be send without caching to browser!

### 2.15.1 Docker image tags

Tag	Distribution name
<code>alpine</code>	Alpine (rolling release)
<code>alpine-3</code>	<i>deprecated</i>
<code>ubuntu-12.04</code>	precise
<code>ubuntu-14.04</code>	trusty (LTS)
<code>ubuntu-15.04</code>	vivid
<code>ubuntu-15.10</code>	wily
<code>ubuntu-16.04</code>	xenial (LTS)
<code>debian-7</code>	wheezy
<code>debian-8</code>	jessie
<code>debian-9</code>	stretch
<code>centos-7</code>	

### 2.15.2 Environment variables

#### Base environment variables

Environment variable	Description	Default
<code>LOG_STDOUT</code>	Destination of daemon output	<i>empty</i> (stdout)
<code>LOG_STDERR</code>	Destination of daemon errors	<i>empty</i> (stdout)
<code>SERVICE_CRON_OPTS</code>	cron daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_OPTS</code>	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_DNSMASQ_USER</code>	dnsmasq effective user	root
<code>SERVICE_POSTFIX_OPTS</code>	postfix daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SSH_OPTS</code>	ssh daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_OPTS</code>	supervisor daemon arguments	<i>empty</i> (when syslog is used)
<code>SERVICE_SUPERVISOR_USER</code>	supervisor effective user	root
<code>SERVICE_SYSLOG_OPTS</code>	syslog daemon arguments	<i>empty</i> (when syslog is used)

## Web development environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
WEB_PHP_TIMEOUT	Webserver PHP request timeout	600
WEB_NO_CACHE_PATTERN	Regex of files which should be delivered by webserver as non cacheable to browser	\. (css js gif png jpg svg json xml)\$

## 2.15.3 Customization

### Ngix customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/nginx/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/nginx/vhost.common.conf` can be used.

Any `*.conf` files inside these direcories will be included either global or the vhost section.

## 2.15.4 Docker image layout

### Ngix layout

File/Directory	Description
<code>/opt/docker/etc/nginx/conf.d</code>	Main global configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/conf.d/10-php.conf</code>	PHP cgi configuration
<code>/opt/docker/etc/nginx/ssl</code>	SSL configuration directory for certifications and keys
<code>/opt/docker/etc/nginx//ssl/server.crt</code>	Example SSL certification (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.csr</code>	Example SSL certification request (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.key</code>	Example SSL key (*.vm)
<code>/opt/docker/etc/nginx/vhost.common.d</code>	Vhost configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/vhost.common.d/10-location-root.conf</code>	Redirect requests to DOCUMENT_INDEX
<code>/opt/docker/etc/nginx/vhost.common.d/10-php.conf</code>	PHP cgi configuration for vhost
<code>/opt/docker/etc/nginx/global.conf</code>	Global nginx configuration
<code>/opt/docker/etc/nginx/main.conf</code>	Main Nginx configuration
<code>/opt/docker/etc/nginx/php.conf</code>	Deprecated PHP configuration
<code>/opt/docker/etc/nginx/vhost.common.conf</code>	Deprecated vhost common include
<code>/opt/docker/etc/nginx/vhost.conf</code>	Vhost configuration
<code>/opt/docker/etc/nginx/vhost.ssl.conf</code>	Vhost SSL configuration
<code>/opt/docker/etc/supervisor.d/nginx.conf</code>	Supervisord configuration file for Nginx

## PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.16 webdevops/php

The php images are based on webdevops/base-app with PHP cli and PHP-FPM. PHP-FPM is running on port 9000.

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.16.1 Docker image tags

Tag	Distribution name	PHP Version
5.6	<i>customized official php image</i>	PHP 5.6
7.0	<i>customized official php image</i>	PHP 7.0
7.1	<i>customized official php image</i>	PHP 7.1
7.2	<i>customized official php image</i>	PHP 7.2
7.3	<i>customized official php image</i>	PHP 7.3
alpine	<i>link to alpine-php7</i>	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	<i>deprecated</i>	PHP 5.6
alpine-3-php7	<i>deprecated</i>	PHP 7.x
ubuntu-12.04	precise	PHP 5.3
ubuntu-14.04	trusty (LTS)	PHP 5.5
ubuntu-15.04	vivid	PHP 5.6
ubuntu-15.10	wily	PHP 5.6
ubuntu-16.04	xenial (LTS)	PHP 7.0
debian-7	wheezy	PHP 5.4
debian-8	jessie	PHP 5.6
debian-8-php7	jessie with dotdeb	PHP 7.x (via sury)
debian-9	stretch	PHP 7.0
centos-7		PHP 5.4
centos-7-php56		PHP 5.6



## 2.16.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

### PHP modules

As we build our images containing almost every PHP module and having it activated by default, you might want to deactivate some.

You can specify a comma-separated list of unwanted modules as dynamic env variable `PHP_DISMOD`, e.g. `PHP_DISMOD=ioncube,redis`.

### PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dynamic env variable which will set `memory_limit = 256M` as php setting.

Environment variable	Description	Default
<code>php.{setting-key}</code>	Sets the <code>{setting-key}</code> as php setting	
PHP_DATE_TIMEZONE	<code>date.timezone</code>	UTC
PHP_DISPLAY_ERRORS	<code>display_errors</code>	0
PHP_MEMORY_LIMIT	<code>memory_limit</code>	512M
PHP_MAX_EXECUTION_TIME	<code>max_execution_time</code>	300
PHP_POST_MAX_SIZE	<code>post_max_size</code>	50M
PHP_UPLOAD_MAX_FILESIZE	<code>upload_max_filesize</code>	50M
PHP_OPCACHE_MEMORY_CONSUMPTION	<code>opcache.memory_consumption</code>	256
PHP_OPCACHE_MAX_ACCELERATED_FILES	<code>opcache.max_accelerated_files</code>	7963
PHP_OPCACHE_VALIDATE_TIMESTAMPS	<code>opcache.validate_timestamps</code>	default
PHP_OPCACHE_REVALIDATE_FREQ	<code>opcache.revalidate_freq</code>	default
PHP_OPCACHE_INTERNED_STRINGS_BUFFER	<code>opcache.interned_strings_buffer</code>	16

### PHP FPM variables

You can specify eg. `fpm.pool.pm.max_requests=1000` as dynamic env variable which will set `pm.max_requests = 1000` as fpm pool setting. The prefix `fpm.pool` is for pool settings and `fpm.global` for global master process settings.

Environment variable	Description	Default
<code>fpm.global.{setting-key}</code>	Sets the {setting-key} as fpm global setting for the master process	
<code>fpm.pool.{setting-key}</code>	Sets the {setting-key} as fpm pool setting	
<code>FPM_PROCESS_MAX</code>	<code>process.max</code>	distribution default
<code>FPM_PM_MAX_CHILDREN</code>	<code>pm.max_children</code>	distribution default
<code>FPM_PM_START_SERVERS</code>	<code>pm.start_servers</code>	distribution default
<code>FPM_PM_MIN_SPARE_SERVERS</code>	<code>pm.min_spare_servers</code>	distribution default
<code>FPM_PM_MAX_SPARE_SERVERS</code>	<code>pm.max_spare_servers</code>	distribution default
<code>FPM_PROCESS_IDLE_TIMEOUT</code>	<code>pm.process_idle_timeout</code>	distribution default
<code>FPM_MAX_REQUESTS</code>	<code>pm.max_requests</code>	distribution default
<code>FPM_REQUEST_TERMINATE_TIMEOUT</code>	<code>request_terminate_timeout</code>	distribution default
<code>FPM_RLIMIT_FILES</code>	<code>rlimit_files</code>	distribution default
<code>FPM_RLIMIT_CORE</code>	<code>rlimit_core</code>	distribution default

### 2.16.3 Customization

#### PHP customization

For customization a placeholder `/opt/docker/etc/php/php.ini` is available which will be loaded as last configuration file. All settings can be overwritten in this ini file.

Either use `COPY` inside your Dockerfile to overwrite this file or use `RUN echo memory_limit = 128 M >> /opt/docker/etc/php/php.ini` to set specific `php.ini` values.

## 2.16.4 Docker image layout

### PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.17 webdevops/php-apache

This image extends `webdevops/php` with a apache daemon which is running on port 80 and 443

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.17.1 Docker image tags

Tag	Distribution name	PHP Version
5.6	<i>customized official php image</i>	PHP 5.6
7.0	<i>customized official php image</i>	PHP 7.0
7.1	<i>customized official php image</i>	PHP 7.1
7.2	<i>customized official php image</i>	PHP 7.2
7.3	<i>customized official php image</i>	PHP 7.3
alpine	<i>link to alpine-php7</i>	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	<i>deprecated</i>	PHP 5.6
alpine-3-php7	<i>deprecated</i>	PHP 7.x
ubuntu-12.04	<b>precise</b>	PHP 5.3
ubuntu-14.04	<b>trusty (LTS)</b>	PHP 5.5
ubuntu-15.04	<b>vivid</b>	PHP 5.6
ubuntu-15.10	<b>wily</b>	PHP 5.6
ubuntu-16.04	<b>xenial (LTS)</b>	PHP 7.0
debian-7	<b>wheezy</b>	PHP 5.4
debian-8	<b>jessie</b>	PHP 5.6
debian-8-php7	<b>jessie with dotdeb</b>	PHP 7.x (via sury)
debian-9	<b>stretch</b>	PHP 7.0
centos-7		PHP 5.4
centos-7-php56		PHP 5.6

## 2.17.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

### Web environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
SERVICE_PHPFPM_OPTS	PHP-FPM command arguments	<i>empty</i> (when php fpm is used)
SERVICE_APACHE_OPTS	Apache command arguments	<i>empty</i> (when apache is used)
SERVICE_NGINX_OPTS	Nginx command arguments	<i>empty</i> (when nginx is used)

### PHP modules

As we build our images containing almost every PHP module and having it activated by default, you might want to deactivate some.

You can specify a comma-separated list of unwanted modules as dynamic env variable `PHP_DISMOD`, e.g. `PHP_DISMOD=ioncube,redis`.

### PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dynamic env variable which will set `memory_limit = 256M` as php setting.

Environment variable	Description	Default
php.{setting-key}	Sets the {setting-key} as php setting	
PHP_DATE_TIMEZONE	date.timezone	UTC
PHP_DISPLAY_ERRORS	display_errors	0
PHP_MEMORY_LIMIT	memory_limit	512M
PHP_MAX_EXECUTION_TIME	max_execution_time	300
PHP_POST_MAX_SIZE	post_max_size	50M
PHP_UPLOAD_MAX_FILESIZE	upload_max_filesize	50M
PHP_OPCACHE_MEMORY_CONSUMPTION	opcache.memory_consumption	256
PHP_OPCACHE_MAX_ACCELERATED_FILES	opcache.max_accelerated_files	7963
PHP_OPCACHE_VALIDATE_TIMESTAMPS	opcache.validate_timestamps	default
PHP_OPCACHE_REVALIDATE_FREQ	opcache.revalidate_freq	default
PHP_OPCACHE_INTERNED_STRINGS_BUFFER	opcache.interned_strings_buffer	16

### PHP FPM variables

You can specify eg. `fpm.pool.pm.max_requests=1000` as dynamic env variable which will set `pm.max_requests = 1000` as fpm pool setting. The prefix `fpm.pool` is for pool settings and `fpm.global` for global master process settings.

Environment variable	Description	Default
fpm.global. {setting-key}	Sets the {setting-key} as fpm global setting for the master process	
fpm.pool. {setting-key}	Sets the {setting-key} as fpm pool setting	
FPM_PROCESS_MAX	process.max	distribution default
FPM_PM_MAX_CHILDREN	pm.max_children	distribution default
FPM_PM_START_SERVERS	pm.start_servers	distribution default
FPM_PM_MIN_SPARE_SERVERS	pm.min_spare_servers	distribution default
FPM_PM_MAX_SPARE_SERVERS	pm.max_spare_servers	distribution default
FPM_PROCESS_IDLE_TIMEOUT	pm.process_idle_timeout	distribution default
FPM_MAX_REQUESTS	pm.max_requests	distribution default
FPM_REQUEST_TERMINATE_TIMEOUT	request_terminate_timeout	distribution default
FPM_RLIMIT_FILES	rlimit_files	distribution default
FPM_RLIMIT_CORE	rlimit_core	distribution default

## 2.17.3 Customization

### PHP customization

For customization a placeholder `/opt/docker/etc/php/php.ini` is available which will be loaded as last configuration file. All settings can be overwritten in this ini file.

Either use `COPY` inside your Dockerfile to overwrite this file or use `RUN echo memory_limit = 128 M >> /opt/docker/etc/php/php.ini` to set specific php.ini values.

### Apache customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/httpd/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/httpd/vhost.common.d` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.17.4 Docker image layout

### Apache layout

File/Directory	Description
<code>/opt/docker/etc/httpd/conf.d</code>	Main global configuration directory (automatically included files)
<code>/opt/docker/etc/httpd/conf.d/10-php.conf</code>	PHP cgi configuration
<code>/opt/docker/etc/httpd/conf.d/10-error-document.conf</code>	Error document configuration
<code>/opt/docker/etc/httpd/conf.d/10-log.conf</code>	Log configuration
<code>/opt/docker/etc/httpd/conf.d/10-server.conf</code>	Basic server configuration
<code>/opt/docker/etc/httpd/ssl</code>	SSL configuration directory for certifications and keys
<code>/opt/docker/etc/httpd/ssl/server.crt</code>	Example SSL certification (*.vm)
<code>/opt/docker/etc/httpd/ssl/server.csr</code>	Example SSL certification request (*.vm)
<code>/opt/docker/etc/httpd/ssl/server.key</code>	Example SSL key (*.vm)
<code>/opt/docker/etc/httpd/vhost.common.d</code>	Vhost configuration directory (automatically included files)
<code>/opt/docker/etc/httpd/vhost.common.d/01-boilerplate.conf</code>	Placeholder configuration file (prevent include errors for Apache 2.2)
<code>/opt/docker/etc/httpd/global.conf</code>	Global httpd configuration
<code>/opt/docker/etc/httpd/main.conf</code>	Main httpd configuration
<code>/opt/docker/etc/httpd/php.conf</code>	Deprecated PHP configuration
<code>/opt/docker/etc/httpd/vhost.common.conf</code>	Deprecated vhost common include
<code>/opt/docker/etc/httpd/vhost.conf</code>	Vhost configuration
<code>/opt/docker/etc/httpd/vhost.ssl.conf</code>	Vhost SSL configuration
<code>/opt/docker/etc/supervisor.d/httpd.conf</code>	Supervisord configuration file for Apache HTTPD

## PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.18 webdevops/php-apache-dev

These image extends `webdevops/php-dev` with a apache daemon which is running on port 80 and 443

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

**Attention:** PHP is running in development mode. Zend Opcode Cache is set to to revalidate files for best development experience.

---

**Tip:** PHP-FPM is accessible by public port 9000

---

## 2.18.1 Docker image tags

Tag	Distribution name	PHP Version
5.6	<i>customized official php image</i>	PHP 5.6
7.0	<i>customized official php image</i>	PHP 7.0
7.1	<i>customized official php image</i>	PHP 7.1
7.2	<i>customized official php image</i>	PHP 7.2
7.3	<i>customized official php image</i>	PHP 7.3
alpine	<i>link to alpine-php7</i>	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	<i>deprecated</i>	PHP 5.6
alpine-3-php7	<i>deprecated</i>	PHP 7.x
ubuntu-12.04	precise	PHP 5.3
ubuntu-14.04	trusty (LTS)	PHP 5.5
ubuntu-15.04	vivid	PHP 5.6
ubuntu-15.10	wily	PHP 5.6
ubuntu-16.04	xenial (LTS)	PHP 7.0
debian-7	wheezy	PHP 5.4
debian-8	jessie	PHP 5.6
debian-8-php7	jessie with dotdeb	PHP 7.x (via sury)
debian-9	stretch	PHP 7.0
centos-7		PHP 5.4
centos-7-php56		PHP 5.6

## 2.18.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>



## PHP modules

As we build our images containing almost every PHP module and having it activated by default, you might want to deactivate some.

You can specify a comma-separated list of unwanted modules as dynamic env variable `PHP_DISMOD`, e.g. `PHP_DISMOD=ioncube,redis`.

## PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dynamic env variable which will set `memory_limit = 256M` as php setting.

Environment variable	Description	Default
<code>php.{setting-key}</code>	Sets the <code>{setting-key}</code> as php setting	
<code>PHP_DATE_TIMEZONE</code>	<code>date.timezone</code>	UTC
<code>PHP_DISPLAY_ERRORS</code>	<code>display_errors</code>	0
<code>PHP_MEMORY_LIMIT</code>	<code>memory_limit</code>	512M
<code>PHP_MAX_EXECUTION_TIME</code>	<code>max_execution_time</code>	300
<code>PHP_POST_MAX_SIZE</code>	<code>post_max_size</code>	50M
<code>PHP_UPLOAD_MAX_FILESIZE</code>	<code>upload_max_filesize</code>	50M
<code>PHP_OPCACHE_MEMORY_CONSUMPTION</code>	<code>opcache.memory_consumption</code>	256
<code>PHP_OPCACHE_MAX_ACCELERATED_FILES</code>	<code>opcache.max_accelerated_files</code>	7963
<code>PHP_OPCACHE_VALIDATE_TIMESTAMPS</code>	<code>opcache.validate_timestamps</code>	default
<code>PHP_OPCACHE_REVALIDATE_FREQ</code>	<code>opcache.revalidate_freq</code>	default
<code>PHP_OPCACHE_INTERNED_STRINGS_BUFFER</code>	<code>opcache.interned_strings_buffer</code>	16

## PHP FPM variables

You can specify eg. `fpm.pool.pm.max_requests=1000` as dynamic env variable which will set `pm.max_requests = 1000` as fpm pool setting. The prefix `fpm.pool` is for pool settings and `fpm.global` for global master process settings.

Environment variable	Description	Default
fpm.global. {setting-key}	Sets the {setting-key} as fpm global setting for the master process	
fpm.pool. {setting-key}	Sets the {setting-key} as fpm pool setting	
FPM_PROCESS_MAX	process.max	distribution default
FPM_PM_MAX_CHILDREN	pm.max_children	distribution default
FPM_PM_START_SERVERS	pm.start_servers	distribution default
FPM_PM_MIN_SPARE_SERVERS	pm.min_spare_servers	distribution default
FPM_PM_MAX_SPARE_SERVERS	pm.max_spare_servers	distribution default
FPM_PROCESS_IDLE_TIMEOUT	pm.process_idle_timeout	distribution default
FPM_MAX_REQUESTS	pm.max_requests	distribution default
FPM_REQUEST_TERMINATE_TIMEOUT	request_terminate_timeout	distribution default
FPM_RLIMIT_FILES	rlimit_files	distribution default
FPM_RLIMIT_CORE	rlimit_core	distribution default

## PHP development environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
WEB_NO_CACHE_PATTERN	RegExp of files which should be delivered by webserver as non cacheable to browser	\.(css js gif png jpg svg json xml)
PHP_DEBUGGER	Specifies which php debugger should be active	xdebug (eg. xdebug, blackfire or none)
XDEBUG_REMOTE_AUTOSTART	php.ini value for xdebug.remote_autostart	none
XDEBUG_REMOTE_CONNECT_BACK	php.ini value for xdebug.remote_connect_back	none
XDEBUG_REMOTE_HOST	php.ini value for xdebug.remote_host	none
XDEBUG_REMOTE_PORT	php.ini value for xdebug.remote_port	none
XDEBUG_PROFILER_ENABLE	php.ini value for xdebug.profiler_enable	none
XDEBUG_PROFILER_ENABLE_TRIGGER	php.ini value for xdebug.profiler_enable_trigger	none
XDEBUG_PROFILER_ENABLE_TRIGGER_VALUE	php.ini value for xdebug.profiler_enable_trigger_value	none
XDEBUG_PROFILER_OUTPUT_DIR	php.ini value for xdebug.profiler_output_dir	none
XDEBUG_PROFILER_OUTPUT_NAME	php.ini value for xdebug.profiler_output_name	none
BLACKFIRE_SERVER_ID	php.ini value for blackfire.server_id	none
BLACKFIRE_SERVER_TOKEN	php.ini value for blackfire.server_token	none
SERVICE_BLACKFIRE_AGENT_ARGS	Blackfire agent command arguments	empty

## 2.18.3 Customization

### PHP customization

For customization a placeholder `/opt/docker/etc/php/php.ini` is available which will be loaded as last configuration file. All settings can be overwritten in this ini file.

Either use `COPY` inside your Dockerfile to overwrite this file or use `RUN echo memory_limit = 128 M >> /opt/docker/etc/php/php.ini` to set specific php.ini values.

### Apache customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/httpd/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/httpd/vhost.common.d` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.18.4 Docker image layout

### Apache layout

File/Directory	Description
/opt/docker/etc/httpd/conf.d	Main global configuration directory (automatically included files)
/opt/docker/etc/httpd/conf.d/10-php.conf	PHP cgi configuration
/opt/docker/etc/httpd/conf.d/10-error-document.conf	Error document configuration
/opt/docker/etc/httpd/conf.d/10-log.conf	Log configuration
/opt/docker/etc/httpd/conf.d/10-server.conf	Basic server configuration
/opt/docker/etc/httpd/ssl	SSL configuration directory for certifications and keys
/opt/docker/etc/httpd/ssl/server.crt	Example SSL certification (*.vm)
/opt/docker/etc/httpd/ssl/server.csr	Example SSL certification request (*.vm)
/opt/docker/etc/httpd/ssl/server.key	Example SSL key (*.vm)
/opt/docker/etc/httpd/vhost.common.d	Vhost configuration directory (automatically included files)
/opt/docker/etc/httpd/vhost.common.d/01-boilerplate.conf	Placeholder configuration file (prevent include errors for Apache 2.2)
/opt/docker/etc/httpd/global.conf	Global httpd configuration
/opt/docker/etc/httpd/main.conf	Main httpd configuration
/opt/docker/etc/httpd/php.conf	Deprecated PHP configuration
/opt/docker/etc/httpd/vhost.common.conf	Deprecated vhost common include
/opt/docker/etc/httpd/vhost.conf	Vhost configuration
/opt/docker/etc/httpd/vhost.ssl.conf	Vhost SSL configuration
/opt/docker/etc/supervisor.d/httpd.conf	Supervisord configuration file for Apache HTTPD

### PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.19 webdevops/php-dev

These image extends `webdevops/php` with `xdebug` and development tools

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

**Attention:** PHP is running in development mode. Zend Opcode Cache is set to to revalidate files for best development experience.

**Tip:** PHP-FPM is accessible by public port 9000

### 2.19.1 Docker image tags

Tag	Distribution name	PHP Version
5.6	<i>customized official php image</i>	PHP 5.6
7.0	<i>customized official php image</i>	PHP 7.0
7.1	<i>customized official php image</i>	PHP 7.1
7.2	<i>customized official php image</i>	PHP 7.2
7.3	<i>customized official php image</i>	PHP 7.3
alpine	<i>link to alpine-php7</i>	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	<i>deprecated</i>	PHP 5.6
alpine-3-php7	<i>deprecated</i>	PHP 7.x
ubuntu-12.04	precise	PHP 5.3
ubuntu-14.04	trusty (LTS)	PHP 5.5
ubuntu-15.04	vivid	PHP 5.6
ubuntu-15.10	wily	PHP 5.6
ubuntu-16.04	xenial (LTS)	PHP 7.0
debian-7	wheezy	PHP 5.4
debian-8	jessie	PHP 5.6
debian-8-php7	jessie with dotdeb	PHP 7.x (via sury)
debian-9	stretch	PHP 7.0
centos-7		PHP 5.4
centos-7-php56		PHP 5.6

### 2.19.2 Environment variables

#### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

### PHP modules

As we build our images containing almost every PHP module and having it activated by default, you might want to deactivate some.

You can specify a comma-separated list of unwanted modules as dynamic env variable `PHP_DISMOD`, e.g. `PHP_DISMOD=ioncube,redis`.

### PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dynamic env variable which will set `memory_limit = 256M` as php setting.

Environment variable	Description	Default
<code>php.{setting-key}</code>	Sets the <code>{setting-key}</code> as php setting	
<code>PHP_DATE_TIMEZONE</code>	<code>date.timezone</code>	UTC
<code>PHP_DISPLAY_ERRORS</code>	<code>display_errors</code>	0
<code>PHP_MEMORY_LIMIT</code>	<code>memory_limit</code>	512M
<code>PHP_MAX_EXECUTION_TIME</code>	<code>max_execution_time</code>	300
<code>PHP_POST_MAX_SIZE</code>	<code>post_max_size</code>	50M
<code>PHP_UPLOAD_MAX_FILESIZE</code>	<code>upload_max_filesize</code>	50M
<code>PHP_OPCACHE_MEMORY_CONSUMPTION</code>	<code>opcache.memory_consumption</code>	256
<code>PHP_OPCACHE_MAX_ACCELERATED_FILES</code>	<code>opcache.max_accelerated_files</code>	7963
<code>PHP_OPCACHE_VALIDATE_TIMESTAMPS</code>	<code>opcache.validate_timestamps</code>	default
<code>PHP_OPCACHE_REVALIDATE_FREQ</code>	<code>opcache.revalidate_freq</code>	default
<code>PHP_OPCACHE_INTERNED_STRINGS_BUFFER</code>	<code>opcache.interned_strings_buffer</code>	16

### PHP FPM variables

You can specify eg. `fpm.pool.pm.max_requests=1000` as dyanmic env variable which will sets `pm.max_requests = 1000` as fpm pool setting. The prefix `fpm.pool` is for pool settings and `fpm.global` for global master process settings.

Environment variable	Description	Default
fpm.global. {setting-key}	Sets the {setting-key} as fpm global setting for the master process	
fpm.pool. {setting-key}	Sets the {setting-key} as fpm pool setting	
FPM_PROCESS_MAX	process.max	distribution default
FPM_PM_MAX_CHILDREN	pm.max_children	distribution default
FPM_PM_START_SERVERS	pm.start_servers	distribution default
FPM_PM_MIN_SPARE_SERVERS	pm.min_spare_servers	distribution default
FPM_PM_MAX_SPARE_SERVERS	pm.max_spare_servers	distribution default
FPM_PROCESS_IDLE_TIMEOUT	pm.process_idle_timeout	distribution default
FPM_MAX_REQUESTS	pm.max_requests	distribution default
FPM_REQUEST_TERMINATE_TIMEOUT	request_terminate_timeout	distribution default
FPM_RLIMIT_FILES	rlimit_files	distribution default
FPM_RLIMIT_CORE	rlimit_core	distribution default

PHP development environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
WEB_NO_CACHE_PATTERN	RegExp of files which should be delivered by webserver as non cacheable to browser	\.(css js gif png jpg svg json xml)
PHP_DEBUGGER	Specifies which php debugger should be active	xdebug (eg. xdebug, blackfire or none)
XDEBUG_REMOTE_AUTOSTART	php.ini value for xdebug.remote_autostart	none
XDEBUG_REMOTE_CONNECT_BACK	php.ini value for xdebug.remote_connect_back	none
XDEBUG_REMOTE_HOST	php.ini value for xdebug.remote_host	none
XDEBUG_REMOTE_PORT	php.ini value for xdebug.remote_port	none
XDEBUG_PROFILER_ENABLE	php.ini value for xdebug.profiler_enable	none
XDEBUG_PROFILER_ENABLE_TRIGGER	php.ini value for xdebug.profiler_enable_trigger	none
XDEBUG_PROFILER_ENABLE_TRIGGER_VALUE	php.ini value for xdebug.profiler_enable_trigger_value	none
XDEBUG_PROFILER_OUTPUT_DIR	php.ini value for xdebug.profiler_output_dir	none
XDEBUG_PROFILER_OUTPUT_NAME	php.ini value for xdebug.profiler_output_name	none
BLACKFIRE_SERVER_ID	php.ini value for blackfire.server_id	none
BLACKFIRE_SERVER_TOKEN	php.ini value for blackfire.server_token	none
SERVICE_BLACKFIRE_AGENT_ARGS	Blackfire agent command arguments	empty

2.19.3 Customization

PHP customization

For customization a placeholder /opt/docker/etc/php/php.ini is available which will be loaded as last configuration file. All settings can be overwritten in this ini file.

Either use COPY inside your Dockerfile to overwrite this file or use RUN echo memory\_limit = 128 M >> /opt/docker/etc/php/php.ini to set specific php.ini values.



## 2.19.4 Docker image layout

### PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.20 webdevops/php-nginx

These image extends `webdevops/php` with a `nginx` daemon which is running on port 80 and 443

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.20.1 Docker image tags

Tag	Distribution name	PHP Version
5.6	<i>customized official php image</i>	PHP 5.6
7.0	<i>customized official php image</i>	PHP 7.0
7.1	<i>customized official php image</i>	PHP 7.1
7.2	<i>customized official php image</i>	PHP 7.2
7.3	<i>customized official php image</i>	PHP 7.3
alpine	<i>link to alpine-php7</i>	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	<i>deprecated</i>	PHP 5.6
alpine-3-php7	<i>deprecated</i>	PHP 7.x
ubuntu-12.04	<b>precise</b>	PHP 5.3
ubuntu-14.04	<b>trusty (LTS)</b>	PHP 5.5
ubuntu-15.04	<b>vivid</b>	PHP 5.6
ubuntu-15.10	<b>wily</b>	PHP 5.6
ubuntu-16.04	<b>xenial (LTS)</b>	PHP 7.0
debian-7	<b>wheezy</b>	PHP 5.4
debian-8	<b>jessie</b>	PHP 5.6
debian-8-php7	<b>jessie with dotdeb</b>	PHP 7.x (via sury)
debian-9	<b>stretch</b>	PHP 7.0
centos-7		PHP 5.4
centos-7-php56		PHP 5.6

## 2.20.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

### Web environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
SERVICE_PHPFPM_OPTS	PHP-FPM command arguments	<i>empty</i> (when php fpm is used)
SERVICE_APACHE_OPTS	Apache command arguments	<i>empty</i> (when apache is used)
SERVICE_NGINX_OPTS	Nginx command arguments	<i>empty</i> (when nginx is used)

### PHP modules

As we build our images containing almost every PHP module and having it activated by default, you might want to deactivate some.

You can specify a comma-separated list of unwanted modules as dynamic env variable `PHP_DISMOD`, e.g. `PHP_DISMOD=ioncube,redis`.

### PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dynamic env variable which will set `memory_limit = 256M` as php setting.

Environment variable	Description	Default
php.{setting-key}	Sets the {setting-key} as php setting	
PHP_DATE_TIMEZONE	date.timezone	UTC
PHP_DISPLAY_ERRORS	display_errors	0
PHP_MEMORY_LIMIT	memory_limit	512M
PHP_MAX_EXECUTION_TIME	max_execution_time	300
PHP_POST_MAX_SIZE	post_max_size	50M
PHP_UPLOAD_MAX_FILESIZE	upload_max_filesize	50M
PHP_OPCACHE_MEMORY_CONSUMPTION	opcache.memory_consumption	256
PHP_OPCACHE_MAX_ACCELERATED_FILES	opcache.max_accelerated_files	7963
PHP_OPCACHE_VALIDATE_TIMESTAMPS	opcache.validate_timestamps	default
PHP_OPCACHE_REVALIDATE_FREQ	opcache.revalidate_freq	default
PHP_OPCACHE_INTERNED_STRINGS_BUFFER	opcache.interned_strings_buffer	16

### PHP FPM variables

You can specify eg. `fpm.pool.pm.max_requests=1000` as dyanmic env variable which will sets `pm.max_requests = 1000` as fpm pool setting. The prefix `fpm.pool` is for pool settings and `fpm.global` for global master process settings.

Environment variable	Description	Default
fpm.global. {setting-key}	Sets the {setting-key} as fpm global setting for the master process	
fpm.pool. {setting-key}	Sets the {setting-key} as fpm pool setting	
FPM_PROCESS_MAX	process.max	distribution default
FPM_PM_MAX_CHILDREN	pm.max_children	distribution default
FPM_PM_START_SERVERS	pm.start_servers	distribution default
FPM_PM_MIN_SPARE_SERVERS	pm.min_spare_servers	distribution default
FPM_PM_MAX_SPARE_SERVERS	pm.max_spare_servers	distribution default
FPM_PROCESS_IDLE_TIMEOUT	pm.process_idle_timeout	distribution default
FPM_MAX_REQUESTS	pm.max_requests	distribution default
FPM_REQUEST_TERMINATE_TIMEOUT	request_terminate_timeout	distribution default
FPM_RLIMIT_FILES	rlimit_files	distribution default
FPM_RLIMIT_CORE	rlimit_core	distribution default

## 2.20.3 Customization

### PHP customization

For customization a placeholder `/opt/docker/etc/php/php.ini` is available which will be loaded as last configuration file. All settings can be overwritten in this ini file.

Either use `COPY` inside your Dockerfile to overwrite this file or use `RUN echo memory_limit = 128 M >> /opt/docker/etc/php/php.ini` to set specific php.ini values.

### Nginx customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/nginx/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/nginx/vhost.common.conf` can be used.

Any `*.conf` files inside these direcories will be included either global or the vhost section.

## 2.20.4 Docker image layout

### Nginx layout

File/Directory	Description
<code>/opt/docker/etc/nginx/conf.d</code>	Main global configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/conf.d/10-php.conf</code>	PHP cgi configuration
<code>/opt/docker/etc/nginx/ssl</code>	SSL configuration directory for certifications and keys
<code>/opt/docker/etc/nginx/ssl/server.crt</code>	Example SSL certification (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.csr</code>	Example SSL certification request (*.vm)
<code>/opt/docker/etc/nginx/ssl/server.key</code>	Example SSL key (*.vm)
<code>/opt/docker/etc/nginx/vhost.common.d</code>	Vhost configuration directory (automatically included files)
<code>/opt/docker/etc/nginx/vhost.common.d/10-location-root.conf</code>	Redirect requests to DOCUMENT_INDEX
<code>/opt/docker/etc/nginx/vhost.common.d/10-php.conf</code>	PHP cgi configuration for vhost
<code>/opt/docker/etc/nginx/global.conf</code>	Global nginx configuration
<code>/opt/docker/etc/nginx/main.conf</code>	Main Nginx configuration
<code>/opt/docker/etc/nginx/php.conf</code>	Deprecated PHP configuration
<code>/opt/docker/etc/nginx/vhost.common.conf</code>	Deprecated vhost common include
<code>/opt/docker/etc/nginx/vhost.conf</code>	Vhost configuration
<code>/opt/docker/etc/nginx/vhost.ssl.conf</code>	Vhost SSL configuration
<code>/opt/docker/etc/supervisor.d/nginx.conf</code>	Supervisord configuration file for Nginx

## PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.21 webdevops/php-nginx-dev

These image extends `webdevops/php-dev` with a `nginx` daemon which is running on port 80 and 443

### Uses Supervisord

This image is using `supervisor` and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified `uid`.

**Attention:** PHP is running in development mode. Zend Opcode Cache is set to to revalidate files for best development experience.

---

**Tip:** PHP-FPM is accessible by public port 9000

---

## 2.21.1 Docker image tags

Tag	Distribution name	PHP Version
5.6	<i>customized official php image</i>	PHP 5.6
7.0	<i>customized official php image</i>	PHP 7.0
7.1	<i>customized official php image</i>	PHP 7.1
7.2	<i>customized official php image</i>	PHP 7.2
7.3	<i>customized official php image</i>	PHP 7.3
alpine	<i>link to alpine-php7</i>	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	<i>deprecated</i>	PHP 5.6
alpine-3-php7	<i>deprecated</i>	PHP 7.x
ubuntu-12.04	precise	PHP 5.3
ubuntu-14.04	trusty (LTS)	PHP 5.5
ubuntu-15.04	vivid	PHP 5.6
ubuntu-15.10	wily	PHP 5.6
ubuntu-16.04	xenial (LTS)	PHP 7.0
debian-7	wheezy	PHP 5.4
debian-8	jessie	PHP 5.6
debian-8-php7	jessie with dotdeb	PHP 7.x (via sury)
debian-9	stretch	PHP 7.0
centos-7		PHP 5.4
centos-7-php56		PHP 5.6

## 2.21.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

## PHP modules

As we build our images containing almost every PHP module and having it activated by default, you might want to deactivate some.

You can specify a comma-separated list of unwanted modules as dynamic env variable `PHP_DISMOD`, e.g. `PHP_DISMOD=ioncube,redis`.

## PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dynamic env variable which will set `memory_limit = 256M` as php setting.

Environment variable	Description	Default
<code>php.{setting-key}</code>	Sets the <code>{setting-key}</code> as php setting	
<code>PHP_DATE_TIMEZONE</code>	<code>date.timezone</code>	UTC
<code>PHP_DISPLAY_ERRORS</code>	<code>display_errors</code>	0
<code>PHP_MEMORY_LIMIT</code>	<code>memory_limit</code>	512M
<code>PHP_MAX_EXECUTION_TIME</code>	<code>max_execution_time</code>	300
<code>PHP_POST_MAX_SIZE</code>	<code>post_max_size</code>	50M
<code>PHP_UPLOAD_MAX_FILESIZE</code>	<code>upload_max_filesize</code>	50M
<code>PHP_OPCACHE_MEMORY_CONSUMPTION</code>	<code>opcache.memory_consumption</code>	256
<code>PHP_OPCACHE_MAX_ACCELERATED_FILES</code>	<code>opcache.max_accelerated_files</code>	7963
<code>PHP_OPCACHE_VALIDATE_TIMESTAMPS</code>	<code>opcache.validate_timestamps</code>	default
<code>PHP_OPCACHE_REVALIDATE_FREQ</code>	<code>opcache.revalidate_freq</code>	default
<code>PHP_OPCACHE_INTERNED_STRINGS_BUFFER</code>	<code>opcache.interned_strings_buffer</code>	16

## PHP FPM variables

You can specify eg. `fpm.pool.pm.max_requests=1000` as dynamic env variable which will set `pm.max_requests = 1000` as fpm pool setting. The prefix `fpm.pool` is for pool settings and `fpm.global` for global master process settings.

Environment variable	Description	Default
fpm.global. {setting-key}	Sets the {setting-key} as fpm global setting for the master process	
fpm.pool. {setting-key}	Sets the {setting-key} as fpm pool setting	
FPM_PROCESS_MAX	process.max	distribution default
FPM_PM_MAX_CHILDREN	pm.max_children	distribution default
FPM_PM_START_SERVERS	pm.start_servers	distribution default
FPM_PM_MIN_SPARE_SERVERS	pm.min_spare_servers	distribution default
FPM_PM_MAX_SPARE_SERVERS	pm.max_spare_servers	distribution default
FPM_PROCESS_IDLE_TIMEOUT	pm.process_idle_timeout	distribution default
FPM_MAX_REQUESTS	pm.max_requests	distribution default
FPM_REQUEST_TERMINATE_TIMEOUT	request_terminate_timeout	distribution default
FPM_RLIMIT_FILES	rlimit_files	distribution default
FPM_RLIMIT_CORE	rlimit_core	distribution default



## PHP development environment variables

Environment variable	Description	Default
WEB_DOCUMENT_ROOT	Document root for webserver	/app
WEB_DOCUMENT_INDEX	Index document	index.php
WEB_ALIAS_DOMAIN	Domain aliases	*.vm
WEB_PHP_SOCKET	PHP-FPM socket address	127.0.0.1:9000 (for php-* images)
WEB_NO_CACHE_PATTERN	RegExp of files which should be delivered by webserver as non cacheable to browser	\.(css js gif png jpg svg json xml)
PHP_DEBUGGER	Specifies which php debugger should be active	xdebug (eg. xdebug, blackfire or none)
XDEBUG_REMOTE_AUTOSTART	php.ini value for xdebug.remote_autostart	none
XDEBUG_REMOTE_CONNECT_BACK	php.ini value for xdebug.remote_connect_back	none
XDEBUG_REMOTE_HOST	php.ini value for xdebug.remote_host	none
XDEBUG_REMOTE_PORT	php.ini value for xdebug.remote_port	none
XDEBUG_PROFILER_ENABLE	php.ini value for xdebug.profiler_enable	none
XDEBUG_PROFILER_ENABLE_TRIGGER	php.ini value for xdebug.profiler_enable_trigger	none
XDEBUG_PROFILER_ENABLE_TRIGGER_VALUE	php.ini value for xdebug.profiler_enable_trigger_value	none
XDEBUG_PROFILER_OUTPUT_DIR	php.ini value for xdebug.profiler_output_dir	none
XDEBUG_PROFILER_OUTPUT_NAME	php.ini value for xdebug.profiler_output_name	none
BLACKFIRE_SERVER_ID	php.ini value for blackfire.server_id	none
BLACKFIRE_SERVER_TOKEN	php.ini value for blackfire.server_token	none
SERVICE_BLACKFIRE_AGENT_ARGS	Blackfire agent command arguments	empty

### 2.21.3 Customization

#### PHP customization

For customization a placeholder `/opt/docker/etc/php/php.ini` is available which will be loaded as last configuration file. All settings can be overwritten in this ini file.

Either use `COPY` inside your Dockerfile to overwrite this file or use `RUN echo memory_limit = 128 M >> /opt/docker/etc/php/php.ini` to set specific php.ini values.

#### Nginx customization

This image has two directories for configuration files which will be automatic loaded.

For global configuration options the directory `/opt/docker/etc/nginx/conf.d` can be used. For vhost configuration options the directory `/opt/docker/etc/nginx/vhost.common.conf` can be used.

Any `*.conf` files inside these directories will be included either global or the vhost section.

## 2.21.4 Docker image layout

### Nginx layout

File/Directory	Description
/opt/docker/etc/nginx/conf.d	Main global configuration directory (automatically included files)
/opt/docker/etc/nginx/conf.d/10-php.conf	PHP cgi configuration
/opt/docker/etc/nginx/ssl	SSL configuration directory for certifications and keys
/opt/docker/etc/nginx//ssl/server.crt	Example SSL certification (*.vm)
/opt/docker/etc/nginx/ssl/server.csr	Example SSL certification request (*.vm)
/opt/docker/etc/nginx/ssl/server.key	Example SSL key (*.vm)
/opt/docker/etc/nginx/vhost.common.d	Vhost configuration directory (automatically included files)
/opt/docker/etc/nginx/vhost.common.d/10-location-root.conf	Redirect requests to DOCUMENT_INDEX
/opt/docker/etc/nginx/vhost.common.d/10-php.conf	PHP cgi configuration for vhost
/opt/docker/etc/nginx/global.conf	Global nginx configuration
/opt/docker/etc/nginx/main.conf	Main Nginx configuration
/opt/docker/etc/nginx/php.conf	Deprecated PHP configuration
/opt/docker/etc/nginx/vhost.common.conf	Deprecated vhost common include
/opt/docker/etc/nginx/vhost.conf	Vhost configuration
/opt/docker/etc/nginx/vhost.ssl.conf	Vhost SSL configuration
/opt/docker/etc/supervisor.d/nginx.conf	Supervisord configuration file for Nginx

### PHP layout

File/Directory	Description
/opt/docker/etc/php/php.webdevops.ini	PHP settings from WebDevOps image
/opt/docker/etc/php/php.ini	php.ini for custom settings
/opt/docker/etc/php/fpm/php-fpm.conf	PHP-FPM main configuration file
/opt/docker/etc/php/fpm/pool.d/application.conf	Application PHP-FPM pool configuration file
/opt/docker/etc/supervisor.d/php-fpm.conf	Supervisord configuration file for PHP-FPM

## 2.22 webdevops/postfix

These image extends `webdevops/base` with a postfix daemon which is running on port 25

### Uses Supervisord

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

## 2.22.1 Docker image tags

Tag	Distribution name
alpine	Alpine (rolling release)
alpine-3	<i>deprecated</i>
ubuntu-12.04	precise
ubuntu-14.04	trusty (LTS)
ubuntu-15.04	vivid
ubuntu-15.10	wily
ubuntu-16.04	xenial (LTS)
debian-7	wheezy
debian-8	jessie
debian-9	stretch
centos-7	

## 2.22.2 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

## 2.23 webdevops/samson-deployment

This image extends zendesk/samson and is a webbased deployment service with Ansistrano, Capistrano and PHP Deployer.

The original image is only the webbased deployment system and is extended by:

- Ansible with Ansistrano
- Capistrano
- PHP Deployer

- Magallanes
- git
- rsync
- docker & docker-compose (as client)
- gulp, grunt, bower
- PHP cli & composer

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

## 2.23.1 Environment variables

### Base environment variables

Environment variable	Description	Default
LOG_STDOUT	Destination of daemon output	<i>empty</i> (stdout)
LOG_STDERR	Destination of daemon errors	<i>empty</i> (stdout)
SERVICE_CRON_OPTS	cron daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_OPTS	dnsmasq daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_DNSMASQ_USER	dnsmasq effective user	root
SERVICE_POSTFIX_OPTS	postfix daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SSH_OPTS	ssh daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_OPTS	supervisor daemon arguments	<i>empty</i> (when syslog is used)
SERVICE_SUPERVISOR_USER	supervisor effective user	root
SERVICE_SYSLOG_OPTS	syslog daemon arguments	<i>empty</i> (when syslog is used)

### Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	<i>empty</i>
POSTFIX_RELAYHOST	Postfix upstream relay server	<i>empty</i>

## 2.23.2 Docker image layout

## 2.24 webdevops/sphinx

This image extends `webdevops/bootstrap` and provides a sphinx build system.

### 2.24.1 Docker image tags

Tag	Distribution name
latest	Alpine 3

## 2.24.2 Usage

```
# Build and watches documentation in ./documentation/docs/
docker run -t -i --rm -p 8080:8000 -v "$(pwd)/documentation/docs:/opt/docs"
↳webdevops/sphinx sphinx-autobuild --poll -H 0.0.0.0 /opt/docs html
```

## 2.25 webdevops/ssh

This image extends `webdevops/bootstrap` and provides a standalone ssh server running on port 22.

### 2.25.1 Docker image tags

Tag	Distribution name
latest	Ubuntu 16.04 xenial (LTS)

### 2.25.2 Environment variables

Environment variable	Description	Default
<code>SERVICE_SSH_OPTS</code>	SSH command arguments	<i>empty</i>

## 2.26 webdevops/storage

This image extends `busybox` and provides `/storage` as volume to other containers.

### 2.26.1 Docker image tags

Tag	Distribution name
latest	Busybox

## 2.27 webdevops/typo3-solr

This image extends `solr` and provides a prebuild image for TYPO3 solr service.

### 2.27.1 Docker image tags

Tag	Distribution name
6.3.0	TYPO3 Solr service for EXT:solr 6.3.0

## 2.28 webdevops/varnish

These image extends `webdevops/base` and provides a standalone varnish server running on port 80.

### 2.28.1 Docker image tags

Tag	Distribution name
latest	Alpine 3

### 2.28.2 Environment variables

Environment variable	Description	Default
VARNISH_PORT	Listening port	80
VARNISH_CONFIG	Custom configuration file	<i>empty</i>
VARNISH_STORAGE	Storage cache setting	<code>malloc, 128m</code>
VARNISH_OPTS	Additional varnish command options	<i>empty</i>
VARNISH_BACKEND_HOST	Backend server hostname	<i>empty</i>
VARNISH_BACKEND_PORT	Backend server port	80

## 2.29 webdevops/vsftp

These image extends `webdevops/base` and provides a standalone vsftp server running on port 20 and 21.

### Uses Supervisor

This image is using supervisor and runs the daemon under user `application` (UID 1000; GID 1000) as default. If the container is started under a different user the daemon will be run under the specified uid.

### 2.29.1 Docker image tags

Tag	Distribution name
latest	Ubuntu 18.04 xenial (LTS)

### 2.29.2 Environment variables

Environment variable	Description	Default
FTP_USER	FTP account username	<code>application</code>
FTP_PASSWORD	FTP account password	<code>application</code>
FTP_UID	FTP account uid	1000
FTP_GID	FTP account gid	1000
FTP_PATH	FTP account home path	<code>/data/ftp/</code>
SERVICE_VSFTP_OPTS	VSFTP command arguments	<i>empty</i>

### 3.1 Requirements

- **python** and **PIP**
- **ruby** for serverspec tests

### 3.2 Install dependencies

The building process, we need some python packages as well as ruby rspec and serverspec packages:

```
make requirements
```

### 3.3 Configuration

All commands are using configuration options from `conf/console.yml`.

### 3.4 bin/console tasks

Tip: Most tasks are using arguments as whitelist addon for easier usage.

#### 3.4.1 bin/console docker:build

Build all Dockerfiles found in `docker/` directory. The directory structure defines the naming of the built images (using convention over configuration).

Option	Description	Values
-v	Verbose output	<i>option only</i>
-threads=n	Run in parallized mode (currently multi-process instead of real threads)	<i>numeric values</i> , auto, auto/2, auto*2, auto-2, auto+2
-dry-run	Don't really execute build process	<i>option only</i>
-no-cache	Don't use Docker caching	<i>option only</i>
-retry=n	Retry process multiple times (eg. for networking issues)	<i>numeric values</i>
-whitelist=term	Only build Docker images with <i>term</i> in name	<i>string value</i>
-black-list=term	Don't build Docker images with <i>term</i> in name	<i>string value</i>

### 3.4.2 bin/console docker:push

Push (upload) all built Docker images to registry (using convention over configuration).

Option	Description	Values
-v	Verbose output	<i>option only</i>
-threads=n	Run in parallized mode (currently multi-process instead of real threads)	<i>numeric values</i> , auto, auto/2, auto*2, auto-2, auto+2
-dry-run	Don't really execute build process	<i>option only</i>
-retry=n	Retry process multiple times (eg. for networking issues)	<i>numeric values</i>
-whitelist=term	Only build Docker images with <i>term</i> in name	<i>string value</i>
-black-list=term	Don't build Docker images with <i>term</i> in name	<i>string value</i>

### 3.4.3 bin/console docker:pull

Pull (download) all built Docker images to registry (using convention over configuration).

Option	Description	Values
-v	Verbose output	<i>option only</i>
-threads=n	Run in parallized mode (currently multi-process instead of real threads)	<i>numeric values</i> , auto, auto/2, auto*2, auto-2, auto+2
-dry-run	Don't really execute build process	<i>option only</i>
-retry=n	Retry process multiple times (eg. for networking issues)	<i>numeric values</i>
-whitelist=term	Only build Docker images with <i>term</i> in name	<i>string value</i>
-black-list=term	Don't build Docker images with <i>term</i> in name	<i>string value</i>

### 3.4.4 bin/console docker:exec

Execute argument as command inside all docker images.

eg. `bin/console docker:exec --whitelist php -- 'php -v'`

Tip: Separate the docker image command arguments from the console commands with two dashes.



Option	Description	Values
-v	Verbose output	<i>option only</i>
-dry-run	Don't really execute build process	<i>option only</i>
-whitelist=term	Only build Docker images with <i>term</i> in name	<i>string value</i>
-blacklist=term	Don't build Docker images with <i>term</i> in name	<i>string value</i>

### 3.4.5 bin/console test:testinfra

Test built images with `testinfra` (python module), spec files are inside `tests/testinfra`

Option	Description	Values
-v	Verbose output	<i>option only</i>
-threads=n	Run in parallized mode (currently multi-process instead of real threads)	<i>numeric values</i> , auto, auto/2, auto*2, auto-2, auto+2
-dry-run	Don't really execute build process	<i>option only</i>
-whitelist=term	Only build Docker images with <i>term</i> in name	<i>string value</i>
-blacklist=term	Don't build Docker images with <i>term</i> in name	<i>string value</i>

### 3.4.6 bin/console test:serverspec

Test built images with `serverspec` (python module), spec files are inside `tests/serverspec`

Option	Description	Values
-threads=n	Run in parallized mode (currently multi-process instead of real threads)	<i>numeric values</i> , auto, auto/2, auto*2, auto-2, auto+2
-dry-run	Don't really execute build process	<i>option only</i>
-retry=n	Retry process multiple times (eg. for networking issues)	<i>numeric values</i>
-whitelist=term	Only build Docker images with <i>term</i> in name	<i>string value</i>
-blacklist=term	Don't build Docker images with <i>term</i> in name	<i>string value</i>

### 3.4.7 bin/console generate:graph

Generates Docker images dependency graph using graphviz.

### 3.4.8 bin/console generate:dockerfile

Generate Dockerfiles from `Dockerfile.jinja2` templates.

Configuration is stored inside `conf/diagram.yml`.

### 3.4.9 bin/console generate:provision

Generate provision (common configuration files) and deploy them to the specified Dockerfile directories.

Configuration is stored inside `conf/provision.yml`.



This section is not done yet!

## 4.1 Docker image tools

### 4.1.1 docker-service

For enabling or disabling services run *docker-service enable* or *docker-service disable* inside your Dockerfile:

```
RUN docker-service enable ssh
```

This task will also trigger an auto installation if the daemon is not installed.

Available services are specified inside */opt/docker/etc/supervisor.d/*.

Common services are:

- cron
- dnsmasq
- postfix
- ssh
- syslog

### 4.1.2 docker-cronjob

For adding cronjobs the *docker-cronjob* script can be used in your Dockerfile:

```
RUN docker-cronjob '* * * * * application /app/cron.php`
```

Because this comand is run in shell mode make sure you add appropriate quotes to disable wildcard matching.

### 4.1.3 docker-provision

The *docker-provision* script controls the ansible provision system. See provision for more details.

(Will be replaced in future)

### 4.1.4 go-replace

Simple but powerful search&replace and template processing tool for manipulating files inside Docker:

```
# normal search&replace
go-replace -s VIRTUAL_HOST -r "$VIRTUAL_HOST" daemon.conf

# or with template
go-replace --mode=template daemon.conf.tmpl:daemon.conf
```

For more informations see documentation inside [go-replace repository](#).

## 4.2 Provisioning

---

**Important:** Provision system is only available in Docker images which are based on `webdevops/base!`

---

### 4.2.1 Custom entrypoint scripts

Shell scripts (\*.sh) for container startup can be placed inside following directories:

- /entrypoint.d/
- /opt/docker/provision/entrypoint.d/

These files (\*.sh) will be executed automatically.

### 4.2.2 Provision Events

Provision event/tag	Description
bootstrap	Run on Docker image creation (only run once)
build	Run on Docker image build
onbuild	Run on Docker image ONBUILD
entrypoint	Run on Docker image entrypoint execution (only here Environment variables are available)

**Attention:** Try to avoid entrypoint provision tasks because it delays startup time.

### 4.2.3 Shell script provision

For each provision event there is a directory for shell scripts:

- /opt/docker/provision/bootstrap.d/

- /opt/docker/provision/build.d/
- /opt/docker/provision/onbuild.d/
- /opt/docker/provision/entrypoint.d/

For customization just add your shell scripts into these directories for the simple shell script provision system.

#### 4.2.4 Ansible provision

For Ansible the provision events are available as tags. The roles are located inside /opt/docker/provision/roles/ and must be registered with the provision system:

```
/opt/docker/bin/provision add --tag bootstrap --role my-own-role
```

Multiple tags can be defined with multiple `--tag` options:

```
/opt/docker/bin/provision add --tag bootstrap --tag build --role my-own-role
```

There is a priority system for roles in which order they should be executed, default priority is 100:

```
## run before
/opt/docker/bin/provision add --tag bootstrap --priority 40 --role my-own-role-first

## run with normal priority
/opt/docker/bin/provision add --tag bootstrap --role my-own-role

## run after
/opt/docker/bin/provision add --tag bootstrap --priority 200 --role my-own-role-last
```

It's also possible to run one role with the provision command:

```
/opt/docker/bin/provision run --tag bootstrap --role my-own-role
```

## 4.3 Supervisor Daemon (Services)

---

**Important:** Supervisor is only available in Docker images which are based on `webdevops/base!`

---

### 4.3.1 Introduction

Supervisor daemon is used to start and supervise more than one process in Docker containers. More about supervisor can be found on supervisor homepage at <http://supervisord.org/>

### 4.3.2 Enable and disable services

For enabling services run `docker-service-enable` or `docker-service-disable` inside your Dockerfile.

eg:

```
RUN docker-service-enable ssh
```

This task will also trigger an auto installation if the daemon is not installed.

### 4.3.3 Configuration

The main supervisor configuration file is located at `/opt/docker/etc/supervisor.conf` and only controls the supervisor daemon itself. All services are configured inside `/opt/docker/etc/supervisor.d/` directory.

Example configuration for hhvm:

```
[group:hhvm]
programs=hhvmd
priority=20

[program:hhvmd]
command = /opt/docker/bin/service.d/hhvm.sh
process_name=%(program_name)s
directory = /var/run/hhvm/
startsecs = 0
autostart = true
autorestart = true
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes=0
stderr_logfile=/dev/stderr
stderr_logfile_maxbytes=0
```

### 4.3.4 Service daemon scripts

For every service there is a small bash script inside `/opt/docker/bin/service.d/` which takes care how to start the service. This script also have a modular task runner which runs files from `/opt/docker/bin/service.d/SERVICE.d/*.sh`.

Example for HHVM:

HHVM's service script is located at `/opt/docker/bin/service.d/hhvm.sh`.

Before running HHVM all scripts found with `/opt/docker/bin/service.d/hhvm.d/*.sh` will be executed.

## 5.1 General images

### 5.1.1 webdevops/...:alpine-3

- doesn't support `locales` (upstream issue)

## 5.2 PHP images

### 5.2.1 webdevops/php...:ubuntu-12.04

- missing `redis`
- missing `apcu`

### 5.2.2 webdevops/php...:debian-7

- missing `redis`
- missing `apcu`

### 5.2.3 webdevops/php...:alpine-3

- missing `mhash`

### 5.2.4 webdevops/php...-dev:alpine-3

- missing `blackfire` (not supported)

### 5.2.5 webdevops/php...-dev:debian-8-php7

- missing xdebug



## CHAPTER 6

---

FAQ

---

Placeholder