Dockerfile Documentation

Release

Florian Tatzel

Jun 26, 2017

Contents

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 progess and we appeciate every help we can get, so feel free to contribute.

Introduction

What are the Dockerfile for?

The Dockerfile repository is reponsible 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/.

CHAPTER 2

Docker images

Fig. 2.1: Docker image hierarchy
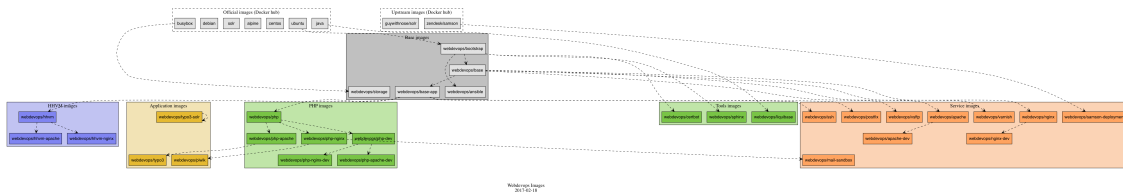
webdevops/ansible

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

Docker image tags

Tag	Distribution name
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	

webdevops/apache

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

Docker image tags

Tag	Distribution name
`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`	

Environment variables

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	*empty* (when php fpm is used)
`SERVICE_APACHE_OPTS`	Apache comamnd arguments	*empty* (when apache is used)
`SERVICE_NGINX_OPTS`	Nginx comamnd arguments	*empty* (when nginx is used)

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 direcories will be included either global or the vhost section.

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

webdevops/apache-dev

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

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

Docker image tags

Tag	Distribution name
`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`	

Environment variables

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

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_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)$`

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 direcories will be included either global or the vhost section.

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

webdevops/base

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

Docker image tags

Tag	Distribution name
`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`	

Environment variables

Base environment variables

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

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"])

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/servide.d/*.sh`	Service scripts if services needs more than just a single command line for startup

Provisioning

With Ansible the provisioning tasks can be easliy done inside the Docker image (eg. for configurations and deployments). There is also a small provision script for registring 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 registred 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)

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!

Docker image tags

Tag	Distribution name
`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`	

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	*empty*
POSTFIX_RELAYHOST	Postfix upstream relay server	*empty*

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.

Docker image tags

Tag	Distribution name
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	

Baselayout scripts

File/Directory	Description
/usr/local/bin/apk-install	**Alpine**: Updates package cache, install packages and clears package cache
/usr/local/bin/apk-upgrade	**Alpine**: Run package upgrade
/usr/local/bin/apt-install	**Debian family**: Updates package cache, install packages and clears package cache
/usr/local/bin/apt-upgrade	**Debian family**: Run package upgrade
/usr/local/bin/yum-install	**RedHat family**: Updates package cache, install packages and clears package cache
/usr/local/bin/yum-upgrade	**RedHat family**: 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 service in Debian)

webdevops/certbot

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

Docker image tags

Tag	Distribution name
`latest`	Alpine 3

webdevops/hhvm

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

Docker image tags

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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

Docker image layout

HHVM layout

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

webdevops/hhvm-apache

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

Docker image tags

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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

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	*empty* (when php fpm is used)
`SERVICE_APACHE_OPTS`	Apache comamnd arguments	*empty* (when apache is used)
`SERVICE_NGINX_OPTS`	Nginx comamnd arguments	*empty* (when nginx is used)

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 direcories will be included either global or the vhost section.

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

HHVM layout

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

webdevops/hhvm-nginx

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

Docker image tags

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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

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	*empty* (when php fpm is used)
`SERVICE_APACHE_OPTS`	Apache comamnd arguments	*empty* (when apache is used)
`SERVICE_NGINX_OPTS`	Nginx comamnd arguments	*empty* (when nginx is used)

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 direcories will be included either global or the vhost section.

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

HHVM layout

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

webdevops/liquibase

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

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

Environment variables

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

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`

POSTGRESL: 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`

webdevops/liquidsoap

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

Docker image tags

Tag	Distribution name
`latest`	Based on *webdevops/base:latest* (Ubuntu)

Environment variables

Base environment variables

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

Liquisoap environment variables

Environment variable	Description	Default
LIQUIDSOAP_USER	Daemon user ID	liquidsoap
LIQUIDSOAP_TELNET	Open telnet (port 1234)	1 (enabled)
LIQUIDSOAP_SCRIPT	Configuration script for liquidsoap	/opt/docker/etc/liquidsoap/default.liq
LIQUIDSOAP_TEMPLATE	Apply template to config script	1/opt/docker/etc/liquidsoap/default.liq
LIQUIDSOAP_STREAM_INPUT	Input stream (eg. icecast)	http://icecast:8000/live
LIQUIDSOAP_PLAYLIST	Default stream when no other stream is active	audio_to_stereo(single('/opt/docker/etc/liquidsoap/default.mp3'))
LIQUIDSOAP_PLAYLIST_DAY	Day input stream/playlist	playlist('/opt/docker/etc/liquidsoap/playlist-day.pls')
LIQUIDSOAP_PLAYLIST_DAY_TIMERANGE	Time range for day playlist	4h-2h
LIQUIDSOAP_PLAYLIST_NIGHT	Night input stream/playlist	playlist('/opt/docker/etc/liquidsoap/playlist-night.pls')
LIQUIDSOAP_PLAYLIST_NIGHT_TIMERANGE	Time range for night playlist	2h-14h
LIQUIDSOAP_OUTPUT	Output stream (eg. icecast)	output.icecast(%mp3(bitrate=128), host='localhost',port=8000, password='secretpassword', mount='liquidsoap-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_META_NAME	Station name	Liquidsoap Docker
LIQUIDSOAP_META_GENRE	Station genre	*empty*
LIQUIDSOAP_META_URL	Station url	*empty*
LIQUIDSOAP_META_DESCRIPTION	Station description	*empty*

webdevops/mail-catcher

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 (round-cube)

Docker image tags

Tag	Distribution name
`latest`	Ubuntu 16.04 xenial (LTS)

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

Mail sandbox environment variables

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

Docker image layout

webdevops/nginx

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

Docker image tags

Tag	Distribution name
`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`	

Environment variables

Base environment variables

Environment variable	Description	Default
`LOG_STDOUT`	Destination of daemon output	*empty* (stdout)
`LOG_STDERR`	Destination of daemon errors	*empty* (stdout)
`SERVICE_CRON_OPTS`	cron daemon arguments	*empty* (when syslog is used)
`SERVICE_DNSMASQ_OPTS`	dnsmasq daemon arguments	*empty* (when syslog is used)
`SERVICE_DNSMASQ_USER`	dnsmasq effective user	`root`
`SERVICE_POSTFIX_OPTS`	postfix daemon arguments	*empty* (when syslog is used)
`SERVICE_SSH_OPTS`	ssh daemon arguments	*empty* (when syslog is used)
`SERVICE_SUPERVISOR_OPTS`	supervisor daemon arguments	*empty* (when syslog is used)
`SERVICE_SUPERVISOR_USER`	supervisor effective user	`root`
`SERVICE_SYSLOG_OPTS`	syslog daemon arguments	*empty* (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	*empty* (when php fpm is used)
`SERVICE_APACHE_OPTS`	Apache comamnd arguments	*empty* (when apache is used)
`SERVICE_NGINX_OPTS`	Nginx comamnd arguments	*empty* (when nginx is used)

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 direcories will be included either global or the vhost section.

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

webdevops/nginx-dev

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

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

Docker image tags

Tag	Distribution name
`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`	

Environment variables

Base environment variables

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

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 direcories will be included either global or the vhost section.

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

webdevops/php

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

Docker image tags

Tag	Distribution name	PHP Version
`5.6`	*customized official php image*	PHP 5.6
`7.0`	*customized official php image*	PHP 7.0
`7.1`	*customized official php image*	PHP 7.1
`alpine`	*link to alpine-php7*	PHP 7.x
`alpine-php7`		PHP 7.x
`alpine-php5`		PHP 5.6
`alpine-3`	*deprecated*	PHP 5.6
`alpine-3-php7`	*deprecated*	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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dyanmic env variable which will sets `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`

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.

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

webdevops/php-apache

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

Docker image tags

Tag	Distribution name	PHP Version
`5.6`	*customized official php image*	PHP 5.6
`7.0`	*customized official php image*	PHP 7.0
`7.1`	*customized official php image*	PHP 7.1
`alpine`	*link to alpine-php7*	PHP 7.x
`alpine-php7`		PHP 7.x
`alpine-php5`		PHP 5.6
`alpine-3`	*deprecated*	PHP 5.6
`alpine-3-php7`	*deprecated*	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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

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	*empty* (when php fpm is used)
`SERVICE_APACHE_OPTS`	Apache comamnd arguments	*empty* (when apache is used)
`SERVICE_NGINX_OPTS`	Nginx comamnd arguments	*empty* (when nginx is used)

PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dyanmic env variable which will sets `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`

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 direcories will be included either global or the vhost section.

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

webdevops/php-apache-dev

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

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

> **Tip:** PHP-FPM is accessable by public port 9000

Docker image tags

Tag	Distribution name	PHP Version
`5.6`	*customized official php image*	PHP 5.6
`7.0`	*customized official php image*	PHP 7.0
`7.1`	*customized official php image*	PHP 7.1
`alpine`	*link to alpine-php7*	PHP 7.x
`alpine-php7`		PHP 7.x
`alpine-php5`		PHP 5.6
`alpine-3`	*deprecated*	PHP 5.6
`alpine-3-php7`	*deprecated*	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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	*empty*
POSTFIX_RELAYHOST	Postfix upstream relay server	*empty*

PHP.ini variables

You can specify eg. php.memory_limit=256M as dyanmic env variable which will sets 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`

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	*empty* (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`	Blackfire agent command arguments	*empty*

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 direcories will be included either global or the vhost section.

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

webdevops/php-dev

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

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

> **Tip:** PHP-FPM is accessable by public port 9000

Docker image tags

Tag	Distribution name	PHP Version
5.6	*customized official php image*	PHP 5.6
7.0	*customized official php image*	PHP 7.0
7.1	*customized official php image*	PHP 7.1
alpine	*link to alpine-php7*	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	*deprecated*	PHP 5.6
alpine-3-php7	*deprecated*	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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	*empty*
POSTFIX_RELAYHOST	Postfix upstream relay server	*empty*

PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dyanmic env variable which will sets `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`

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	*empty* (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`	Blackfire agent command arguments	*empty*

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.

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

webdevops/php-nginx

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

Docker image tags

Tag	Distribution name	PHP Version
`5.6`	*customized official php image*	PHP 5.6
`7.0`	*customized official php image*	PHP 7.0
`7.1`	*customized official php image*	PHP 7.1
`alpine`	*link to alpine-php7*	PHP 7.x
`alpine-php7`		PHP 7.x
`alpine-php5`		PHP 5.6
`alpine-3`	*deprecated*	PHP 5.6
`alpine-3-php7`	*deprecated*	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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

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	*empty* (when php fpm is used)
`SERVICE_APACHE_OPTS`	Apache comamnd arguments	*empty* (when apache is used)
`SERVICE_NGINX_OPTS`	Nginx comamnd arguments	*empty* (when nginx is used)

PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dyanmic env variable which will sets `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`

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.

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

webdevops/php-nginx-dev

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

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

> **Tip:** PHP-FPM is accessable by public port 9000

Docker image tags

Tag	Distribution name	PHP Version
5.6	*customized official php image*	PHP 5.6
7.0	*customized official php image*	PHP 7.0
7.1	*customized official php image*	PHP 7.1
alpine	*link to alpine-php7*	PHP 7.x
alpine-php7		PHP 7.x
alpine-php5		PHP 5.6
alpine-3	*deprecated*	PHP 5.6
alpine-3-php7	*deprecated*	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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	*empty*
POSTFIX_RELAYHOST	Postfix upstream relay server	*empty*

PHP.ini variables

You can specify eg. `php.memory_limit=256M` as dyanmic env variable which will sets `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`

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	*empty* (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`	Blackfire agent command arguments	*empty*

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.

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

webdevops/postfix

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

Docker image tags

Tag	Distribution name
`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`	

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
`POSTFIX_MYNETWORKS`	Postfix mynetworks address	*empty*
`POSTFIX_RELAYHOST`	Postfix upstream relay server	*empty*

webdevops/samson-deployment

These 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

Environment variables

Base environment variables

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

Base environment variables

Environment variable	Description	Default
POSTFIX_MYNETWORKS	Postfix mynetworks address	*empty*
POSTFIX_RELAYHOST	Postfix upstream relay server	*empty*

Docker image layout

webdevops/sphinx

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

Docker image tags

Tag	Distribution name
latest	Alpine 3

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
```

webdevops/ssh

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

Docker image tags

Tag	Distribution name
`latest`	Ubuntu 16.04 xenial (LTS)

Environment variables

Environment variable	Description	Default
`SERVICE_SSH_OPTS`	SSH command arguments	*empty*

webdevops/storage

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

Docker image tags

Tag	Distribution name
`latest`	Busybox

webdevops/typo3-solr

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

Docker image tags

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

webdevops/varnish

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

Docker image tags

Tag	Distribution name
`latest`	Alpine 3

Environment variables

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

webdevops/vsftp

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

Docker image tags

Tag	Distribution name
latest	Ubuntu 16.04 xenial (LTS)

Environment variables

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

Commands (bin/console)

Requirements

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

Install dependencies

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

```
make requirements
```

Configuration

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

bin/console tasks

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

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

bin/console docker:push

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

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

bin/console docker:pull

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

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

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	*option only*
–dry-run	Don't really execute build process	*option only*
–whitelist=term	Only build Docker images with *term* in name	*string value*
–blacklist=term	Don't build Docker images with *term* in name	*string value*

bin/console test:testinfra

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

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

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)	*numeric values*, auto, auto/2, auto*2, auto-2, auto+2
–dry-run	Don't really execute build process	*option only*
–retry=n	Retry process multiple times (eg. for networking issues)	*numeric values*
–whitelist=term	Only build Docker images with *term* in name	*string value*
–blacklist=term	Don't build Docker images with *term* in name	*string value*

bin/console generate:graph

Generates Docker images dependency graph using graphviz.

bin/console generate:dockerfile

Generate Dockerfiles from `Dockerfile.jinja2` templates.

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

bin/console generate:provision

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

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

Customization

This section is not done yet!

Docker image tools

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

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.

docker-php-setting

Only available on php images!

This scripts sets php.ini setting globaly:

```
RUN docker-php-setting memory_limit 1G

RUN docker-php-setting --raw error_reporting 'E_ALL & ~E_NOTICE & ~E_STRICT & ~E_
↪DEPRECATED'
```

docker-provision

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

(Will be replaced in future)

go-replace

Simple but powerfull 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.

Provisioning

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

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.

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.

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.

Ansible provision

For Ansible the provision events are available as tags. The roles are located inside `/opt/docker/provision/roles/` and must be registred 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 pritory 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
```

Supervisor Daemon (Services)

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

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/

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.

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
```

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.

Known issues

General images

webdevops/...:alpine-3

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

PHP images

webdevops/php...:ubuntu-12.04

- missing redis
- missing apcu

webdevops/php...:debian-7

- missing redis
- missing apcu

webdevops/php...:alpine-3

- missing mhash

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

- missing blackfire (not supported)

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

- missing xdebug

FAQ

Placeholder