

Title: Docker-Compose won't volume my php.ini file

Post Body:

I'm trying to use docker-compose to volume my php.ini file so I can make changes on the fly on my local machine to see how it affects the host machine. Unfortunately the only way I've been able to get the php.ini file into the container is directly during creation in the Dockerfile so far.

Attached is an image of the container running fine with the current settings below.

My Dockerfile is below:

```
FROM ubuntu:14.04 MAINTAINER Joe Astrahan <jastrahan@poolservice.software> VOLUME ['/var/www'] RUN apt-get update && \
```

My docker-compose.yml file is below:

```
version: '2' services:      dblive:      image: mysql:5.5.52      volumes:      - ./db_data_live:/var/lib/mysql
```

I tried following the advice here, [can't update php.ini file in Docker container](#), by creating a custom.ini file and mounting it in that location. I actually did it correctly I think because if you look at my image I attached for phpinfo(), you can see that under additional .ini files parsed my custom.ini is there at the end. I did a test though by setting asp_tags = On instead of Off and I can't. phpinfo() will always show it as off. Refer to my attached image of it showing it off despite loading my config file.

I'm not even sure if its really honoring any of the commands in there at all?

Extra Files Used

Run

```
#!/bin/bash set -e  PHP_ERROR_REPORTING=${PHP_ERROR_REPORTING:-'E_ALL & ~E_DEPRECATED & ~E_NOTICE'} sed -ri 's/^display_errors
```

install_php-5.5.30.sh

```
#!/bin/bash # install dependencies apt-get -y update && \ apt-get install -y \  build-essential \  apache2-dev \  libxml2-
```

My file structure

Accepted Answer:

I found the answer, put a file called custom.php.ini in the config directory (if you are following my directory structure).

Set the volumes like this in my example...

```
volumes:      - ./web:/var/www      - ./config/custom.php.ini:/etc/php5/apache2/conf.d/custom.php.ini
```

By default the scan directory for extra php files will look in the conf.d directory. These files will overwrite the settings of the main php.ini. I tested this with the asp_tag option turning it Off and On. It works fine as long as you do the following below.

The trick to making this work is to use `docker-compose down` instead of `docker-compose kill`

This removes the containers and their cache files. PHP only loads the configuration file once at bootup and the other files so changing the php.ini or the custom file after requires this docker-compose down to force the change.

Highest Rated Answer:

If you are using something like **wodby** (docker4php or docker4drupal) or **lando** or trying to find an answer "why php.ini doesn't work" (like me), these tools are using their own way to pass configuration into php

<https://github.com/wodby/php#php-and-php-fpm-configuration>

I am posting this answer here because I wasted 2 hours to find an answer and I came into this question from google. I want to save some time to others.

Mounting of custom.php.ini will not help.