Easily changing code in a docker container

Last updated by | Eric Woodworth | Jul 29, 2020 at 8:29 AM CDT

Problem Statement

I'm writing celery code on windows 10 and celery hates windows. It's possible to get it working with a command like this:

```
celery worker --app=app.app --pool=solo
```

but it doesn't work if you want to have multiple workers. Since I wanted to run multiple workers I wanted to run in containers.

Using Containers

Containers are cool - that's for sure, but every time I change code I don't want to go through a full container build. There are other solutions for this but the way I did it pretty simple

```
  1) I put the dockerfile and docker-compose in the same folder
  2) I run docker-compose up from that folder
  3) If I change something about the build the container gets rebuilt, otherwise the cached version is used
  4) I edit my code in E:\virutalenvs\celery\code which the docker container mounts in /code
  5) This lets me execute the code in the container in /code and if the code changes I just restart docker-compo
```

docker-compose.yml

```
version: '3'
services:
    add:
        build: .
        user: celery
        command: ls /code
        command: bash -c "cd /code && celery -A add worker -Q add"
        volumes:
            - E:\virtualenvs\celery\code:/code

    subtract:
        build: .
        user: celery
        command: ls /code
        command: bash -c "cd /code && celery -A subtract worker -Q subtract"
        volumes:
            - E:\virtualenvs\celery\code:/code

    divide:
        build: .
        user: celery
        command: ls /code
        command: bash -c "cd /code && celery -A divide worker -Q divide"
        volumes:
            - E:\virtualenvs\celery\code:/code

    multiply:
        build: .
        user: celery
        command: ls /code
        command: bash -c "cd /code && celery -A multiply worker -Q multiply"
        volumes:
            - E:\virtualenvs\celery\code:/code

    custom:
        build: .
        user: celery
        command: ls /code
        command: bash -c "cd /code && celery -A custom_thing worker -Q custom"
        volumes:
            - E:\virtualenvs\celery\code:/code
```

dockerfile

```
FROM ubuntu:20.04
RUN apt update -qq && apt -y install python3-pip
RUN mkdir /code
RUN pip3 install redis
RUN pip3 install celery
RUN pip3 install pika
RUN pip3 install pex
RUN useradd celery
RUN chown celery:celery /code
```