Dockerize your Pyramid Application

This article will walk through how to dockerize your Pyramid application. Pyramid is a lightweight web framework made from the developers who created the Pylons web application framework. Reddit, O'Reilly, SurveyMonkey, and SourceForge (among others) all use Pylons Project software.

The Docker Hub image

The Pylons Project does not offer an official image for Pyramid, so we'll create our own Dockerfile, starting from Ubuntu. While you can always use an un-official image, its generally recommended to create your own so you know what your Docker image is made up of.

Setting up

We need to set up a basic practice app and Dockerfile.

Our basic app

Start with creating a new directory; let's call it `pyramid_hello_world`:

```
mkdir pyramid_hello_world
```

Copy and paste the contents below to a new file `pyramid_hello_world/app.py`:

```
# pyramid_hello_world/app.py

from wsgiref.simple_server import make_server
from pyramid.config import Configurator
from pyramid.response import Response

def hello_world(request):
    print('Request inbound!')
    return Response('Docker works with Pyramid!')


if __name__ == '__main__':
    config = Configurator()
    config.add_route('hello', '/')
    config.add_view(hello_world, route_name='hello')
    app = config.make_wsgi_app()
    server = make_server('0.0.0.0', 6543, app)
    server.serve_forever()
```

Now we need to include Pyramid in the `requirements.txt` file:

```
pyramid==1.7
```

Pyramid Dockerfile

We're starting from Linux instead of using the Python repository as our base, as its more clear in which Python version is being installed (what `apt` installs on Ubuntu or Debian, or `yum` installs on Red Hat and CentOS). You always have the option to build off the `python` image instead.

```
FROM ubuntu:16.04

RUN apt-get update -y && \
    apt-get install -y python-pip python-dev && \
    pip install --upgrade pip setuptools

# We copy this file first to leverage docker cache
COPY ./requirements.txt /app/requirements.txt

WORKDIR /app

RUN pip install -r requirements.txt
```

```
COPY . /app

ENTRYPOINT [ "python" ]

CMD [ "app.py" ]
```

Let’s go over some of these Docker instructions:

1. `COPY` copies files from the first parameter (the source, `.`) to the destination parameter (in this case, `/app`)
2. `WORKDIR` sets your working directory; you may set and reset this as often as you need
3. `ENTRYPOINT` sets the container to run as an executable

Build the image

Now we’re ready to see if the Dockerfile builds successfully:

```
docker build -t pyramid-tutorial:latest .
```

After the image has been built, we can then run the container:

```
docker run -d -p 6543:6543 pyramid-tutorial
```

Further information

Ensure you’re using the correct port. Pyramid by defualt uses port 6543. Take a look at Binding Docker Ports for more information.

Do not use more than one `ENTRYPOINT` in your Dockerfile (Docker only uses the last one).

Next: Docker Compose with Django Apps
Create repeatable Django environments using Docker Compose.