Conda Environments with Docker

Chad Lagore Follow
Mar 9, 2018 · 2 min read

Conda is a nice choice for a package manager in Python. The primary reason I end up using it over pip or pipenv, is to install Python packages that have *non-Python dependencies (*written in C, Fortran or some other compilable language). Conda does this dependency resolution and source installation quite well. In fact chances are, if your pip installation is failing (for non-permissions reasons), Conda can probably help you.

Using Conda with Docker was a bit strange at first. Ideally the project should make use of an `environment.yml` (a refreshing upgrade from the traditional `requirements.txt`) — but this requires using a Conda environment — so now we'll have a virtual environment within a Docker container. It's not an unwelcome bit of extra isolation, but if a environment exists within the container, it's best that we don't have to ever think about it. I've seen some heroic attempts to make the Conda environment transparent to the user.

Happily the solution is easier than you might expect, here's the `Dockerfile` stub to get you started.

```
FROM continuumio/miniconda3

RUN conda create -n env python=3.6
RUN echo "source activate env" > ~/.bashrc
ENV PATH /opt/conda/envs/env/bin:$PATH
```

Build it,

```
$ docker build -t condatest .
Sending build context to Docker daemon  160.8kB
Step 1/3 : FROM continuumio/miniconda
```

```
 ---> 53447bc91028
Step 2/3 : RUN conda create -n env python=3.5
 ---> Using cache
 ---> 60042252066c
Step 3/3 : RUN echo "source activate env" > ~/.bashrc
 ---> Running in 48bf43a954f6
Removing intermediate container 48bf43a954f6
 ---> 0f902c2d6431
Successfully built 0f902c2d6431
Successfully tagged condatest:latest
$ docker run -it condatest
(env) root@8b7a59b17d5f:/#
```

Notice that we're immediately running within the virtual environment. Heres the stub for an environment created from your `environment.yml` .

```
FROM continuumio/miniconda3

ADD environment.yml /tmp/environment.yml
RUN conda env create -f /tmp/environment.yml

# Pull the environment name out of the environment.yml
RUN echo "source activate $(head -1 /tmp/environment.yml | cut -d' '
-f2)" > ~/.bashrc
ENV PATH /opt/conda/envs/$(head -1 /tmp/environment.yml | cut -d' ' -
f2)/bin:$PATH
```

A slight tweak on the environment activation command lets you use the same environment name that we specified at the top of our `environment.yml` . Hope this helps!

Docker Conda Python Containers Dependencies

About Help Legal