Using docker image for python interpreter in PyCharm

It is possible to use a docker image as the project interpreter in PyCharm. This can be helpful when extractors require dependencies that can be difficult to install (like GDAL). Since a docker image will eventually be created for any extractor, it also makes sense to build this into the development process as early as possible. This also ensures that the local environment will be the same as the docker image deployed elsewhere. The problem of something working locally (but not in docker) or an extractor that runs (but not locally) can be avoid using this method.

Step 1: Build Docker Image (if one does not exist)

In order to be used as a python interpreter, the docker image must at least contain python and contain all currently required dependencies. You will not have to push the image to a remote repository; as long as it is local PyCharm will find it.

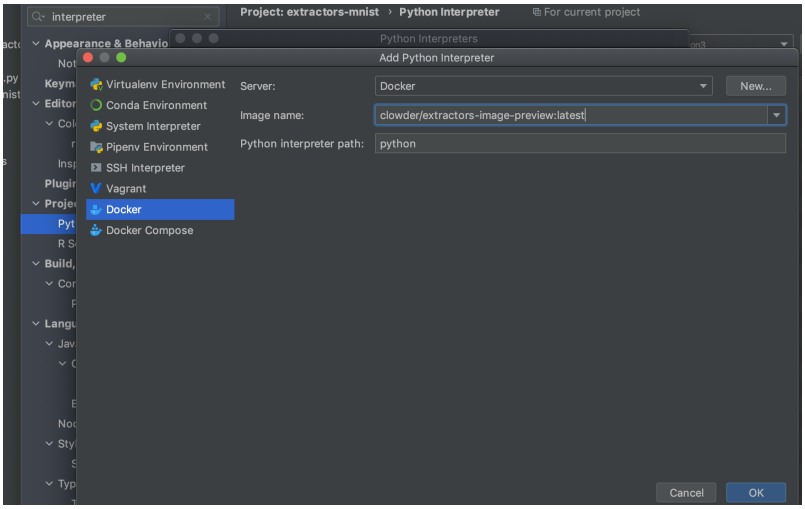
Step 2: Pull docker image if using a remote image

If you would like to use an existing docker image, do a docker pull for that image.

Step 3: Add Docker Image as Python Interpreter in PyCharm

Search for Python Interpreter and select add or manage. Then select 'docker' - you will then see all docker images pulled locally and can select from among them.

When adding the run configuration, make sure to select this for the extractor.



Step 4: Modify environment variable in extractor python file.

```
os.environ['RABBITMQ_URI'] = 'amqp://guest:guest@{your-ip-here}/%2f'
os.environ['CLOWDER_URL'] = 'http://{your-ip-here}:8000/'
os.environ['REGISTRATION_ENDPOINTS'] ='http://{your-ip-here}:8000/api/extractors?key={your-key-here}'
```

Step 5: Modify value for RabbitMQ URL in clowder docker-compose file

```
- RABBITMQ_CLOWDERURL=${RABBITMQ_CLOWDERURL:-http://{your-ip-here:8000}
```

Step 6: How to run:

User docker-compose command to start clowder and its basic dependencies (mongo, rabbitmq, etc.) Once those are started, you should be able to run the extractor in PyCharm, in either run or debug mode. Note that running in docker may affect what the extractor considers the current working directory and path to files.

Step 7: If you want to modify the docker image

If you modify and build a new Docker image, make sure to REMOVE it from the Python Interpreters in PyCharm, and then add it again. This will make sure it refreshes.