Title: Docker and Rabbitmq: ECONNREFUSED between containers

Post Body:

I am trying to setup separate docker containers for rabbitmq and the consumer for the container, i.e., the process that would listen on the queue and perform the necessary tasks. I created the yml file, and the docker file.

I am able to run the yml file, however when I check the docker-compose logs I see where there are ECONNREFUSED errors.

NewUserNotification.js:

```
require('seneca')()     .use('seneca-amqp-transport')     .add('action:new_user_notification’, function(message, done) {
```

error message in docker-compose log:

```
    {'notice':'seneca: Action hook:listen,role:transport,type:amqp failed: connect ECONNREFUSED 127.0.0.1:5672.','code':     '
```

sample docker-compose.yml file:

```
version: '2.1' services:  rabbitmq:     container_name: '4340_rabbitmq'     tty: true     image: rabbitmq:management     ports
```

amqp connection string: (I tried both ways, with and without a user/pass) amqp://username:password@rabbitmq:5672

I added the link attribute to the docker-compose file and referenced the name in the .env file(rabbitmq). I tried to run the NewUserNotification.js file from outside the container and it started fine. What could be causing this problem? Connection string issue? Docker-Compose.yml configuration issue? Other?

Accepted Answer:

Seems the environment variable `AMQP_RECEIVE_URL` is not constructed properly. According to error log the listener is trying to connect to localhost(127.0.0.1) which is not the rabbitmq service container IP. Find the modified configurations for a working sample.

1 docker-compose.yml

```
version: '2.1' services:  rabbitmq:     container_name: '4340_rabbitmq'     tty: true     image: rabbitmq:management     ports
```

2 un.env

```
AMQP_RECEIVE_URL=amqp://guest:guest@rabbitmq:5672
```

Note that I've passed the `AMQP_RECEIVE_URL` as an environment variable to `new_user_notification` service using `env_file` and got rid of the `account` service

3 Dockerfile

```
FROM node:7 WORKDIR /app COPY newusernotification.js /app COPY wait-for-it.sh /app RUN npm install --save seneca RUN npm insta
```

4 `newusernotification.js` use the same file in the question.

5 wait-for-it.sh

Highest Rated Answer:

It is possible that your RabbitMQ service is not fully up, at the time the connection is attempted from the consuming service.

If this is the case, in Docker Compose, you can `wait` for services to come up using a container called dadarek/wait-for-dependencies.

1). Add a new service `waitforrabbit` to your docker-compose.yml

```
waitforrabbit:   image: dadarek/wait-for-dependencies   depends_on:     - rabbitmq    command: rabbitmq:5672
```

2). Include this service in the `depends_on` section of the service that requires RabbitMQ to be up.

```
depends_on:    - waitforrabbit
```

3). Startup compose

```
docker-compose run --rm waitforrabbit docker-compose up -d account new_user_notification
```

Starting compose in this manner will essentially wait for RabbitMQ to be fully up before the connection from the consuming service is made.