Title: Docker port isn't accessible from host

Post Body:

I have a new Spring Boot application that I just finished and am trying to deploy it to Docker. Inside the container the application works fine. It uses ports 9000 for user facing requests and 9100 for administrative tasks like health checks. When I start a docker instance and try to access port 9000 I get the following error:

```
curl: (56) Recv failure: Connection reset by peer
```

After a lot of experimentation (via curl), I confirmed in with several different configurations that the application functions fine inside the container, but when I try to map ports to the host it doesn't connect. I've tried starting it with the following commands. None of them allow me to access the ports from the host.

```
docker run -P=true my-app docker run -p 9000:9000 my-app
```

The workaround

The only approach that works is using the --net host option, but this doesn't allow me to run more than one container on that host.

```
docker run -d --net=host my-app
```

Experiments with ports and expose

I've used various versions of the Dockerfile exposing different ports such as 9000 and 9100 or just 9000. None of that helped. Here's my latest version:

```
FROM ubuntu MAINTAINER redacted  RUN apt-get update RUN apt-get install openjdk-7-jre-headless -y RUN mkdir -p /opt/app  WORKD
```

Hello World works

To make sure I can run a Spring Boot application, I tried Simplest-Spring-Boot-MVC-HelloWorld and it worked fine.

Netstat Results

I've used netstat to do port scans from the host and from the container:

From the host

```
root@my-docker-host:~# nmap 172.17.0.71 -p9000-9200  Starting Nmap 6.40 ( http://nmap.org ) at 2014-11-14 19:19 UTC Nmap scan
```

From the container

```
root@80cf20c0c1fa:/opt/app# nmap 127.0.0.1 -p9000-9200  Starting Nmap 6.40 ( http://nmap.org ) at 2014-11-14 19:20 UTC Nmap sc
```

The container is using Ubuntu The hosts I've replicated this are Centos and Ubuntu.

This SO question seems similar but had very few details and no answers, so I thought I'd try to document my scenario a bit more.

Accepted Answer: None

Highest Rated Answer:

I had a similar problem, in which specifying a host IP address as '127.0.0.1' wouldn't properly forward the port to the host.

Setting the web server's IP to '0.0.0.0' fixes the problem

eg - for my Node app - the following **doesn't** work

```
app.listen(3000, '127.0.0.1')
```

Where as the following **does** work:

```
app.listen(3000, '0.0.0.0')
```

Which I guess means that docker, by default, is exposing 0.0.0.0:containerPort -> local port