Title: Connection to localhost:5432 refused after docker run when port 5432 is clearly open and listening?

Post Body:

I have a ci service (specifically gitlab ci, if it matters) that is docker-compose'ing a postgres instance at `0.0.0.0:5432` which is then needed for maven integration tests I run within a dockerfile.

For some reason, and you can see the stack trace, the dockerfile maven integration tests can't connect to the container open on 5432.

Why is this happening and how do I fix the connection?

FWIW, when I run `docker-compose up` and run the integration tests locally (as opposed to within a dockerfile) everything passes fine.

```
$ docker ps CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
```

```
07-07-2020 20:46:01.634 [main] INFO  com.zaxxer.hikari.HikariDataSource.getConnection - HikariPool-1 - Starting... 07-07-2020
```

will update with answers

> Have you tried checking if you flushed the iptables rules that docker created? if so restart docker service, and double check the iptables rules: iptables -nvL

```
$ iptables -nvL Chain INPUT (policy ACCEPT 0 packets, 0 bytes)  pkts bytes target     prot opt in     out     source
```

> What OS are you running? is SELinux in enforcing mode? getenforce if so check if something got denied: grep -i denied /var/log/audit/audit.log

```
$ cat /etc/*release 3.12.0 NAME="Alpine Linux" ID=alpine VERSION_ID=3.12.0 PRETTY_NAME="Alpine Linux v3.12" HOME_URL="https://
```

> I don't think this is a pg_hba.conf but it does not hurt to check the config.

I'm not sure how to do that?

> # test with telnet in you can mannualy connect: telnet localhost 5432

```
$ telnet localhost 5432 telnet: can't connect to remote host (127.0.0.1): Connection refused
```

> As a last resort while running docker-compose up watch a tcpdump: tcpdump -nni any port 5432 maybe the traffic got redirected somewhere, or localhost is badly defined in /etc/hosts

```
$ tcpdump -nni any port 5432 tcpdump: verbose output suppressed, use -v or -vv for full protocol decode listening on any, link
```

when i try different host

application properties

```
spring.datasource.url=jdbc:postgresql://db:5432/*****************db
```

```
08-07-2020 16:24:57.484 [main] WARN  org.springframework.web.context.support.GenericWebApplicationContext.refresh - Exception
```

Accepted Answer: None

Highest Rated Answer:

It is because if you are using docker compose to expose a service port, you need to connect to the `service name` instead of localhost

https://docs.docker.com/compose/networking/

If the name of your service is `postgres` just point your app to `postgres:5432`