

How do I access a server on localhost with nginx docker container?

Asked 10 years, 3 months ago Modified 10 months ago Viewed 98k times



75

I'm trying to use a dockerized version of nginx as a proxy server for my node (ExpressJS) application. Without any configuration to nginx and publishing port 80 for the container, I am able to see the default nginx landing page. So I know that much is working.



Now I can mount my sites-enabled directory that contains the configuration for `proxy_pass localhost:3000`. I have my node application running locally (not in any Docker container) and I can access it via port 3000 (i.e. `localhost:3000`). However, I would assume that with nginx container running, mapped to port 80, and proxying my localhost:3000, that I would be able to see my very simple (hello world) application. Instead I receive a 502.



Do I need to pass something into docker? Is this likely a nginx configuration error? Here is my nginx configuration:

```
server {  
    listen 0.0.0.0:80;  
    server_name localhost;  
  
    location / {  
        proxy_pass http://localhost:3000;  
    }  
}
```

I have tried using [this question](#) but it did not seem to help. That is unless I'm doing something completely wrong.

node.js

nginx

docker

Share Improve this question Follow

asked Jan 7, 2015 at 1:27



adam-beck

6,019 ● 5 ● 22 ● 35


possible duplicate of [Connect to Local MySQL Server Through Docker?](#) – Bryan Jan 7, 2015 at 11:47

I tried that as well and couldn't seem to get it to work. I know this similar question has been asked many times and even the 502 error I'm getting has been too, but none of those have worked. I will re-check when I am home from work. – adam-beck Jan 7, 2015 at 13:37

DNS ftw, imo. The host has a name ;) – user2105103 Jan 7, 2015 at 19:27

docker on linux, version >= 20.04 can use `host.docker.internal` with a run flag; see stackoverflow.com/a/61424570/2014893 – Robert K. Bell Aug 17, 2020 at 6:28

7 Answers

Sorted by: Highest score (default) 

86



If you're using docker-for-mac 18.03 or newer it auto creates a special DNS entry `host.docker.internal` that dynamically binds to the host inet ip. You can then use the dns name to proxy services running on the host machine from inside a container as a stand-in for `localhost`.

i.e. an nginx config file:

```
server {
    listen 0.0.0.0:80;
    server_name localhost;

    location / {
        proxy_pass http://host.docker.internal:3000;
    }
}
```

Share Improve this answer Follow

answered Jun 28, 2018 at 15:06



[digitaldreamer](#)

55.7k ● 6 ● 35 ● 28

14 Note: `host.docker.internal` does not work on linux. – [21rw](#) Apr 11, 2021 at 3:22

6 Unless you set `extra-hosts` stackoverflow.com/a/43541732/6790541 – [21rw](#) Apr 11, 2021 at 5:10

3 This also worked for me on a Windows 10 host. – [Joe Sadoski](#) Sep 17, 2021 at 15:28



27



You can get your current IP address as shown [here](#):

```
ifconfig en0 | grep inet | grep -v inet6 | awk '{print $2}'
```

Then you can use the `--add-host` flag with `docker run`:

```
docker run --add-host localnode:$(ifconfig en0 | grep inet | grep -v inet6 |
awk '{print $2}') ...
```

In your `proxypass` use `localnode` instead of `localhost`.

Share Improve this answer Follow

edited May 23, 2017 at 12:00

answered Jan 7, 2015 at 7:40



1 This was along the right lines but did not work exactly for me. For one I am working off my laptop so my method of extracting the ip was obviously different. I tried using the same method changing `en0` but kept getting results with `addr:192.xxx.x.xx` so I eventually just hard-coded that in. What utility would I use to strip the beginning `addr:` as providing that will not work with docker. Thank you though. I'm writing this comment for others who come across this problem. – [adam-beck](#) Jan 8, 2015 at 2:34

1 Nevermind, I used `sed` in order to strip the beginning `addr:`. As you can tell I'm not a sysops guy :) – [adam-beck](#) Jan 8, 2015 at 13:35

On my Ubuntu, it's `eth0` instead of `en0`. – [tedyyu](#) Dec 10, 2023 at 10:49



5



Yes. Docker needs to know about your host machine. You can set an alias to that with the `--add-host` switch. On a *nix box to create an alias to a name "localbox", this would be:

```
docker run my_repo/my_image --add-host=localbox:<host_name>
```

On boot2docker it would be:

```
docker run my_repo/my_image --add-host=localbox:192.168.59.3
```

where you should replace "192.168.59.3" with whatever `boot2docker ip` returns.

Then, you should access your host machine always through the alias localbox, so just change your nginx config to:

```
location / {
    proxy_pass http://localbox:3000;
}
```

Share Improve this answer Follow

answered Jan 7, 2015 at 5:14



Eli

39.1k ● 40 ● 155 ● 213



3



On linux, this works for me:

In the docker-compose.yml, mount an entrypoint script into the nginx container:

```
nginx:
  image: nginx:1.19.2
  # ...
```



volumes:

```
- ./nginx-entrpoint.sh:/docker-entrpoint.d/nginx-entrpoint.sh:ro
```

The contents of the entrpoint map a local address to the host local address.

```
apt update
apt install iproute2 -y
echo "`ip route | awk '/default/ { print $3 }'`\tdocker.host.internal" >>
/etc/hosts
```

Then, instead of using `localhost` inside the container, you can use `docker.host.internal`.

Share Improve this answer Follow

answered Sep 15, 2020 at 4:50



Matthias

15.6k ● 9 ● 49 ● 65



0



You can get access a server on `localhost` by docker desktop host networking.

Host networking is also supported on Docker Desktop version 4.29 and later for Mac, Windows, and Linux as a *beta feature*. To enable this feature, navigate to the **Features in development** tab in **Settings**, and then select **Enable host networking**.



The reference is [here](#)

Share Improve this answer Follow

answered Jun 20, 2024 at 6:08



iku

1,144 ● 13 ● 16



-2



I had the same problem. Fixed it by using the local ip address of the docker host, instead of localhost.

So if the local ip address of your docker host in your LAN is 192.168.2.2:

```
location / {
    proxy_pass http://192.168.2.2:3000;
}
```



Of course this solution only works well if you have assigned a static ip to your docker host.

Share Improve this answer Follow

answered Oct 17, 2022 at 9:02



Jakub

123 ● 1 ● 11



-3



And finally, if you are using Nginx as a reverse proxy for multiple services, you can spin all of that with docker-compose. Make sure to expose ports "80:80" only on the Nginx service. Other services you can expose only the service port without mapping to the underlying network like so:

```
web:
  ....
  expose:
    - 8080
nginx:
  ....
  port:
    - "80:80"
```

and then use Nginx configuration proxy_pass <http://service-name:port> You don't need the upstream app part at all

Share Improve this answer Follow

edited Jan 10, 2019 at 9:58

answered Aug 12, 2018 at 7:47



E_net4

30.2k ● 13 ● 114 ● 151



George Mogilevsky

637 ● 5 ● 8

It's not clear what you mean here by "upstream app part" that we won't need. Also this ends up being not very portable if `http://service-name` is your external domain instead of localhost. That was the first alternative I found to localhost, but I don't like it because I can't be sure that my nginx redirects aren't in fact going back out to the internet just to come in on a different port. – [jcairney](#) Oct 8, 2019 at 0:08

Please read the docs, it's a technical term that you should be familiar with.

nginx.org/en/docs/http/nginx_http_upstream_module.html – [George Mogilevsky](#) Oct 8, 2019 at 0:14

It is a technical term I was familiar with, but I wouldn't expect everyone to be, nor was I even sure that's what you were referring to. Also, I do not know why you mentioned it, since nobody else mentioned 'upstream' above. I would not consider the 'upstream' module a beginner feature, so it's not very sporting to say "you should know this". This is stackoverflow, where people come because they can't know everything. – [jcairney](#) Oct 31, 2019 at 19:04

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

node.js

nginx

docker

See similar questions with these tags.

