

DNS resolution in docker containers

Asked 5 years, 2 months ago Modified 2 years, 8 months ago Viewed 116k times



[environment]

25



- CentOS 7
- Docker 19.03.12, build 48a66213fe



I have a failing **cerbot** (let's encrypt client) inside a **docker** container. It looks like `acme-v02.api.letsencrypt.org` is not resolvable **from the container** but is resolvable **from the host** (the most probable cause). I am used to the fact that containers inherit from the host's DNS parameters but, in the case of AWS EC2 instances, there seems to be some subtleties

DNS Manual setting

```
[ec2-user@ip-172-31-32-243 ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search eu-west-2.compute.internal
nameserver 172.31.0.2
```

Based on this and some elements from **AWS console**, I tried to add these address manually

```
docker run --dns 172.31.0.2 --dns 172.65.32.248
```

(I probably did not find the proper DNS)

An inelegant fix

Using the **host's network** solves the problem

```
docker run --network="host"
```

But I just don't quite understand why. Is it a matter of network interface used in `--network="host"`. Here are the one available:

```
[ec2-user@ip-172-31-32-243 ~]$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_code1 state UP
mode DEFAULT group default qlen 1000
    link/ether 0a:d2:81:33:16:f2 brd ff:ff:ff:ff:ff:ff
```

```

3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state
DOWN mode DEFAULT group default
    link/ether 02:42:9c:c8:d4:ba brd ff:ff:ff:ff:ff:ff
8: br-d15fdfe7243b: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue
state DOWN mode DEFAULT group default
    link/ether 02:42:e8:56:81:bf brd ff:ff:ff:ff:ff:ff

```

Would it be because `--network="host"` induce using `eth0` instead of `docker0`? How can I resolve `acme-v02.api.letsencrypt.org` from `docker0`?

Detailed error

```

An unexpected error occurred:
Traceback (most recent call last):
  File "/usr/lib/python3.8/site-packages/urllib3/connection.py", line 156, in
_new_conn
    conn = connection.create_connection(
  File "/usr/lib/python3.8/site-packages/urllib3/util/connection.py", line 61,
in create_connection
    for res in socket.getaddrinfo(host, port, family, socket.SOCK_STREAM):
  File "/usr/lib/python3.8/socket.py", line 918, in getaddrinfo
    for res in _socket.getaddrinfo(host, port, family, type, proto, flags):
socket.gaierror: [Errno -3] Try again

```

During handling of the above exception, another exception occurred:

```

Traceback (most recent call last):
  File "/usr/lib/python3.8/site-packages/urllib3/connectionpool.py", line 665,
in urlopen
    httplib_response = self._make_request(
  File "/usr/lib/python3.8/site-packages/urllib3/connectionpool.py", line 376,
in _make_request
    self._validate_conn(conn)
  File "/usr/lib/python3.8/site-packages/urllib3/connectionpool.py", line 994,
in _validate_conn
    conn.connect()
  File "/usr/lib/python3.8/site-packages/urllib3/connection.py", line 334, in
connect
    conn = self._new_conn()
  File "/usr/lib/python3.8/site-packages/urllib3/connection.py", line 168, in
_new_conn
    raise NewConnectionError(
urllib3.exceptions.NewConnectionError:
<urllib3.connection.VerifiedHTTPSConnection object at 0x7f55dcc4a130>: Failed
to establish a new connection: [Errno -3] Try again

```

During handling of the above exception, another exception occurred:

```

Traceback (most recent call last):
  File "/usr/lib/python3.8/site-packages/requests/adapters.py", line 439, in
send
    resp = conn.urlopen(
  File "/usr/lib/python3.8/site-packages/urllib3/connectionpool.py", line 719,
in urlopen
    retries = retries.increment(
  File "/usr/lib/python3.8/site-packages/urllib3/util/retry.py", line 436, in
increment
    raise MaxRetryError(_pool, url, error or ResponseError(cause))

```

```
urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='acme-v02.api.letsencrypt.org', port=443): Max retries exceeded with url: /directory (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f55dcc4a130>: Failed to establish a new connection: [Errno -3] Try again'))
```

During handling of the above exception, another exception occurred:

```
requests.exceptions.ConnectionError: HTTPSConnectionPool(host='acme-v02.api.letsencrypt.org', port=443): Max retries exceeded with url: /directory (Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at 0x7f55dcc4a130>: Failed to establish a new connection: [Errno -3] Try again'))
```

Please see the logfiles [in /var/log/letsencrypt](#) for more details.

docker

dns

certbot

Share Improve this question Follow

edited Jul 18, 2020 at 12:54



Chris Williams

35.6k ● 4 ● 46 ● 79

asked Jul 18, 2020 at 12:52



zar3bski

3,213 ● 8 ● 32 ● 69

+1 to [Suhas NM] (stackoverflow.com/a/62988037/7878132). For more detailed explanation, you can refer to : 1. stackoverflow.com/a/50730336/7878132 2. github.com/moby/moby/pull/41022 – KoRa Aug 21, 2022 at 10:00 ✎

2 Answers

Sorted by: Highest score (default) ⇅



Looks like DNS resolution inside docker is not working properly.

30

For linux systems, DNS resolution happens using `/etc/resolv.conf` file, check this file inside your container, if it has invalid DNS, then your container won't be able to resolve hostnames.



Docker uses a property set in `/etc/docker/daemon.json` file(on host) for populating entries in `/etc/resolv.conf` inside container.



update the value of this property in `/etc/docker/daemon.json` file in host machine:

```
{
  "dns": ["8.8.8.8"]
}
```

Note: This change requires restarting docker to take effect, also existing containers have to be removed and created again.

Restarting Docker:

```
sudo systemctl restart docker
```

removing containers:

```
sudo docker stop <container-name/id>
sudo docker rm <container-name/id>
```

You can add multiple DNS server ip addresses, separate each one with a comma - check [here](#) to get a list of DNS server ip addresses

Cheers!!

Share Improve this answer Follow

answered Jul 20, 2020 at 3:03



Suhas NM

1,090 ● 9 ● 11

! Sign up to request clarification or add additional context in comments.



1 Comment ▼

Add a comment



Minh Thiện Over a year ago

Need to be in same docker network too, see stackoverflow.com/a/35691865

▲ 0

Reply



The answer of Suhas has solved a problem when I'm trying to launch a "python:latest" container image and installing a requirements.txt list inside of it.

2



This trick is very useful because some images you cannot write anything to resolv.conf because you may not have privileges to add new dependent root layers there.



example of error:



```
" > [6/6] RUN pip3 install -r requirements.txt:
#0 14.69 WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None,
status=None)) after connection broken by
'NewConnectionError('<pip._vendor.urllib3.connection.HTTPSConnection object at
0x7fa212596110>: Failed to establish a new connection: [Errno -3] Temporary failure in
name resolution')': /simple/flask/ "
```

If you guys get some compose stack to run something like Flask server or any python xxxx.py program you must add those DNS entries in your host's docker daemon.json file. Create it if it not exists.

create a DNS entry in your docker daemon:

```
sudo cat <<EOF > /etc/docker/daemon.json
{
  "dns": ["8.8.8.8"]
}
EOF
```

check you firewall status first (RHEL based - disable for testing or add an rule for your compose port)

```
systemctl status firewalld
```

restart host's docker service

```
systemctl restart docker
systemctl daemon-reload
```

then run you stack

```
docker-compose up -d --build
```

Share Improve this answer Follow

edited Jan 22, 2023 at 8:53

answered Jan 22, 2023 at 8:46



[Alan Aguinaga](#)

21 ● 4

Comments

Add a comment

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

[docker](#) [dns](#) [certbot](#)

See similar questions with these tags.