nginx resolver -- dns

Asked 7 years ago Modified 3 years, 4 months ago Viewed 72k times

30

Please excuse a very beginner question.

I'm having trouble understanding the nginx 'resolver' parameter and how it works. I have read the documentation, searched tutorials and posts (using keywords like resolver, nginx, and dns), and I'm still not sure how to apply resolver.

http://nginx.org/en/docs/http/ngx_http_core_module.html#resolver

"Configures name servers used to resolve names of upstream servers into addresses...."

- By this definition, it seems to be simply doing the nameserver's job. `resolver ns1.myhost.com ns2.myhost.com;` But the examples point to an internal/private IP address.

"An address can be specified as a domain name or IP address, and an optional port...."

- This implies that I could `resolver example.com www.example.com;` (or `resolver 12.34.56.78;`) but again, I see no such examples in the documentation.

As a practical example, let's say — purely hypothetically :) — that I'm building a simple web server with a couple of server blocks on it.

Do I set 'resolver' to the IP of the server itself? Or an internal IP in the server's LAN? The documentation seems to suggest an internal IP (127.x.x.x or 10.x.x.x) — but how to set/determine what that IP is?

nginx dns resolver

Share Improve this question Follow

edited Jun 26, 2019 at 16:54
guntbert
536 ● 6 ● 19

asked Apr 4, 2018 at 2:11
Geekomancer
489 ● 1 ● 4 ● 12

3 Answers

Sorted by: Highest score (default)

31

Resolve means which DNS server nginx should refer to when it has to resolve an external url. If you have a config like below

```
location / {
    proxy_pass http://www.example.com/abc/def;
}
```

Now by default `nginx` will pick your resolver from the host `/etc/resolv.conf`, but it may not be what you need. If you want to use the Google DNS resolver, then you will update your nginx config like below:

```
location / {
    resolver 8.8.8.8;
    proxy_pass http://www.example.com/abc/def;
}
```

If you are using a local DNS resolver to route within your local network, then you may use something like below:

```
location / {
    resolver 192.168.11.10;
    proxy_pass http://machineabc/abc/def;
}
```

Share Improve this answer Follow

edited Dec 8, 2021 at 18:22

Josh Correia
4,363 ● 3 ● 42 ● 63

answered Apr 4, 2018 at 3:30

Tarun Lalwani
147k ● 11 ● 213 ● 276

3 Note: If the host part of `proxy_pass` is defined statically (i.e. the host doesn't need to be looked up in a variable) as in the example `proxy_pass http://machineabc/abc/def`, then the host will be resolved **once** and that time will be during NGNIX startup/reload. See the answers on "How to force nginx to resolve DNS (of a dynamic hostname) everytime when doing proxy_pass?" on Server Fault for more details.
– Anon Aug 8, 2019 at 2:50

10

"Resolver" parameter defines the location of DNS server that nginx must use in order to resolve the IP of the URL passed under proxy_pass;

As explained by Tarun, by default nginx will pick your resolver from the host /etc/resolv.conf and once resolved, it will cache the IP. Resolver is mostly used in two cases:
1. Either in a private network, to resolve the IP's that exist in your network.
2. Or used at a place where the IP of your proxy_pass or upstream location changes very frequently and you cannot rely upon nginx cached IP.

In the example you specified, the resolver will be the IP of the DNS server that could resolve your location. This could be either of :

1) 127.0.0.1 : If the web server itself is a DNS server, for this you need to setup DNS server on port 53(default) of this server.

2) x.x.x.x : The IP of the DNS server hosted in, either in your private network or any public DNS server if your URL's are publicly accessible. One may use 8.8.8.8 (Google's public DNS server).

3) You specified 10.x.x.x : Assuming that you were referring the AWS documentation. If not, in general, 10.x.x.x again needs to be a DNS server IP, which in case of AWS is 10.0.0.2. AWS reserves a few IP's of its VPC's and the second IP x.x.x.2 is reserved for DNS server. Note that in case your VPC is not 10.0.0.0/16, this IP will change accordingly. Eg: Lets say your VPC is 10.192.0.0/16, then you will be using 10.192.0.2 as resolver.

For above ref to https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html

https://www.jethrocarr.com/2013/11/02/nginx-reverse-proxies-and-dns-resolution/

Share Improve this answer Follow

answered Jun 26, 2018 at 21:35

Nishith Kulshrestha
146 ● 1 ● 4

-2

There is another way to make this, if you want to set manually the resolution, without using external tools like bind9 or dnsmasq

```
location / {
    set $upstream 12.34.56.78;  # desired IP resolution
    proxy_pass  http://$upstream:8080; # desired port
    proxy_set_header Host example.com; # desired host
}
```

Share Improve this answer Follow

answered Oct 21, 2020 at 10:37

Hayashi
1

2 This is irrelevant for the asked question, and better done with `upstream backend { server 12.34.56.78; }` because rewrite module directives (set) are known to produce unexpected behavior in more complex cases. – temoto Nov 4, 2020 at 12:31

See similar questions with these tags.