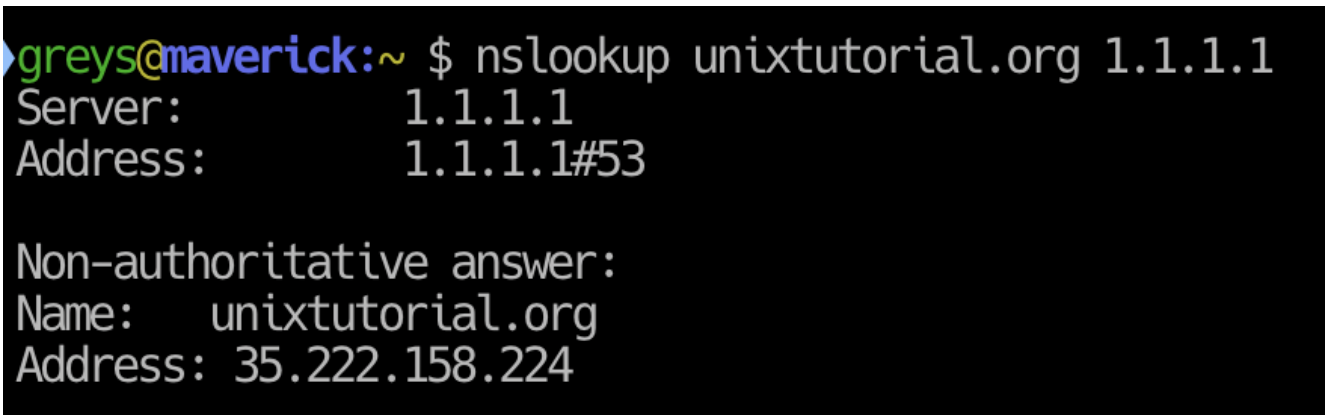
5 Ways to Use nslookup



Even after a few years of blogging I still manage to find must-have and must-know commands that I barely mentioned here. **nslookup** is one such command – it's an indispensable network troubleshooting tool when it comes to troubleshooting DNS issues.

Even after a few years of blogging I still manage to find must-have and must-know commands that I barely mentioned here. **nslookup** is one such command – it's an indispensable network troubleshooting tool when it comes to troubleshooting DNS issues.

Get IP address for a Hostname

This is the most common nslookup command usage: you know a hostname or a domain name, and you want to get an IP address.

```
greys@maverick:~ $ nslookup unixtutorial.org
Server:         192.168.1.1
Address:    192.168.1.1#53
```

```
Non-authoritative answer:
Name:    unixtutorial.org
Address: 35.222.158.224
```

Get hostname from an IP address

This is the opposite of the previous step, but it’s not always returning the opposite result. You see, if your domain name points to an IP address this IP address is not always pointing back to your domain:

```
greys@maverick:~ $ nslookup 35.222.158.224
Server:         192.168.1.1
Address:    192.168.1.1#53

Non-authoritative answer:
224.158.222.35.in-addr.arpa                 name    =
224.158.222.35.bc.googleusercontent.com.
```

Confirm Name Servers for a Domain

When something about DNS resolution doesn’t make sense, I usually go back to basics: confirm the actual Name Servers that provide name resolution for the hostname or domain name.

For this, we need to make nslookup query specifically the NS records for a given hostname:

```
greys@maverick:~ $ nslookup -query=ns unixtutorial.org
Server:         192.168.1.1
```

```
Address:    192.168.1.1#53

Non-authoritative answer:
unixtutorial.org    nameserver = alec.ns.cloudflare.com.
unixtutorial.org    nameserver = beth.ns.cloudflare.com.
```

The output confirms that NS servers called **alec** and **beth** from **Cloudflare** are managing my unixtutorial.org domain.

Confirm Mail Servers for a Domain

Another really cool thing is that **nslookup** can get you MX recourds – they are pointing to the mail (SMTP) servers accepting emails on behalf of the domain:

```
greys@maverick:~ $ nslookup -query=mx unixtutorial.org
Server:         192.168.1.1
Address:    192.168.1.1#53

Non-authoritative answer:
unixtutorial.org    mail exchanger = 1 aspmx.l.google.com.
unixtutorial.org    mail exchanger = 5 alt1.aspmx.l.google.com.
```

This output confirms that my email is handled by Google, because I'm using the gSuite.

Query a Specific DNS Server Using

nslookup

You may have noticed from examples that nslookup always reports my home office server (192.168.1.1) that it uses for making DNS queries.

Sometimes you want to know if other DNS servers resolve the name correctly, and so this example shows you how. Mind you, this must be either your local network (corporate) DNS server or a public NS server – otherwise your request may be rejected (because by default NS servers only resolve their own domains, not all the domains for the rest of the internet).

Here's how I can resolve using Cloudflare's public DNS service:

```
greys@maverick:~ $ nslookup unixtutorial.org 1.1.1.1
Server:         1.1.1.1
Address:    1.1.1.1#53

Non-authoritative answer:
Name:    unixtutorial.org
Address: 35.222.158.224
```

… or Google Public DNS:

```
greys@maverick:~ $ nslookup unixtutorial.org 8.8.8.8
Server:         8.8.8.8
Address:    8.8.8.8#53

Non-authoritative answer:
Name:    unixtutorial.org
```

```
Address: 35.222.158.224
```

That’s it for today! There’s quite a few things nslookup is good for, but I think I’ll need to explain some DNS (Domain Name Service) basics first.

See Also

- Unix Commands
- nslookup
- traceroute
- ip command
- Test TCP connectivity with curl
- Show open ports with lsof
- lsof command