nslookup: Here’s how the useful DNS check works

As the central directory service for network addresses, the domain name system (DNS) plays an important role in the success of the Internet and World Wide Web. The **web of globally distributed DNS servicers** (also called name servers) does the important work of ensuring that the names of the different **network participants and applications**, such as *example.org*, are converted into numeric and machine-readable IP addresses (and vice versa). In this way, it guarantees that the correct computer or desired website is always reached via the corresponding IP, even if you are unaware of it.

In some situations (when there are issues with the name resolution, for example), it is helpful to look behind the scenes and **research the IP address linked to a domain name**, or the **domain name linked to an IP address**. A useful tool for this is the program **nslookup**, which Windows, macOS, and Linux all feature as a standard installation.

What is nslookup?

nslookup is a simple but very practical command-line tool, which is principally used to find the **IP address** that corresponds to a host, or the domain name that corresponds to an IP address (a process called “Reverse DNS Lookup”). nslookup allows itself to be used in the command-line of the operating system in question; Windows users start the service via the **command prompt**, and Unix users via the **terminal window**. Additionally, there are now a number of services that make it possible to use nslookup online.

nslookup retrieves the relevant address information directly from the DNS cache of name servers, a process which can be achieved through two different modes that the user can choose from. In the **noninteractive mode**, the tool inspects the **resource records** (which is what the address

entries in the DNS cache are called) that are stored in the local name server, in a standard way. This mode is especially well suited for simple queries, for which a single domain entry needs to be looked up. When you want to use a different DNS server for the research and complete more complex search processes, you need the **interactive mode**, in which the command-line program needs to be started separately at first.

1. How to find the A record of a domain.

You can use this command to see how many A records are there and see the IP Addresses of each one.

Command line:

$ nslookup example.com

2. How to check the NS records of a domain.

By checking the NS records, you can see which is the authoritative server for a specific domain.

Command line:

$nslookup -type=ns example.com

3. How to query the SOA record of a domain.

With this one, you can see the start of authority and get information about the zone.

Command line:

$nslookup -type=soa example.com

4. How to find the MX records responsible for the email exchange.

Here we are checking the MX records of the mail servers. You can see if all the mail servers are working well.

Command line:

$ nslookup -query=mx example.com

5. How to find all of the available DNS records of a domain.

This lookup has a large scope. Here we want to see all the available DNS records. After seeing all of them, we can do specific lookups for different types of DNS records.

Command line:

$ nslookup -type=any example.com

6. How to check the using of a specific DNS Server.

Apart from checking DNS records, you can use the Nslookup to review a particular DNS server and how it works. You can check if it is active or if it responds on time.
Command line:
$ nslookup example.com ns1.nsexample.com

7. How to check the Reverse DNS Lookup.

Many times you check the A records to see the IPs of a domain, but sometimes you need to verify if an IP address is related to a specific domain. For that purpose, we need a reverse DNS lookup.
Command line:
$ nslookup 10.20.30.40

8. Nslookup command to change the port number for the connection.

You can check the connection through different ports. This can help you to see if there are open ports that you don't use. Later you can close them for security reasons.
Command line:
$ nslookup -port=56 example.com

9.How to change the timeout interval for a reply.

You can manually choose the timeout time in seconds. You can increase it to give more time for the server to respond. You can also shorter it to see which servers can respond quicker.
Command line:
$ nslookup -timeout=20 example.com

10. How to enable debug mode.

Debug mode provides important and detailed information both for the question and for the received answer.

Command line:
$ nslookup -debug example.com

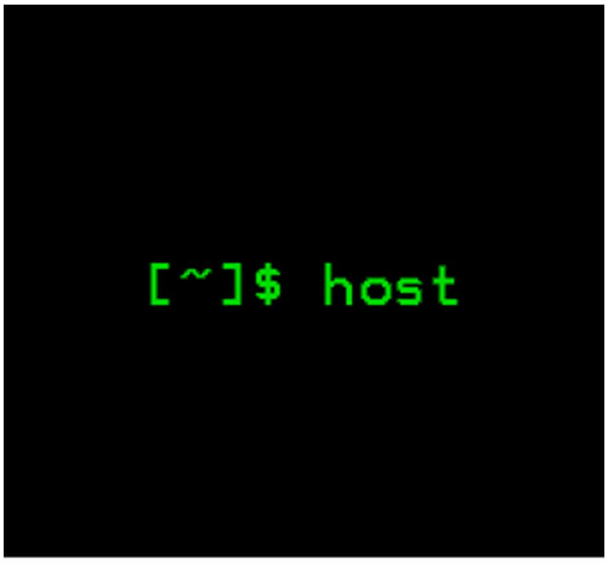
Nslookup is one of the popular command-line software for DNS probing.

Notes:

Authoritative answer – This is the answer that originates from the DNS Server which has the information about the zone file.

Non-authoritative answer – When a nameserver is not in the list for the domain you did a lookup on.

Different port – By default, the DNS servers use port 53.



On Unix-like operating systems, the **host** command is a DNS lookup utility, finding the IP address of a domain name. It also performs reverse lookups, finding the domain name associated with an IP address.

To find out the IP address of **linux-bible.com**, type *host linux-bible.com*:

```
susel:~ # host linux-bible.com
linux-bible.com has address 198.57.241.163
linux-bible.com mail is handled by 0 linux-bible.com.
```

To find out the hostname of the host with the IP address of **208.117.229.34**, use the following command:

```
susel:~ # host 208.117.229.34
34.229.117.208.in-addr.arpa domain name pointer cache.google.com.
```

To print the **SOA record information**, use the -*C* option. A SOA (Start of Authority) record contains basic properties of the domain and the zone that the domain is in.

```
susel:~ # host -C google.com
Nameserver 216.239.38.10:
        google.com has SOA record ns1.google.com. dns-admin.google.com. 20140218
00 7200 1800 1209600 300
Nameserver 216.239.36.10:
        google.com has SOA record ns1.google.com. dns-admin.google.com. 20140218
00 7200 1800 1209600 300
Nameserver 216.239.34.10:
        google.com has SOA record ns1.google.com. dns-admin.google.com. 20140218
00 7200 1800 1209600 300
Nameserver 216.239.32.10:
        google.com has SOA record ns1.google.com. dns-admin.google.com. 20140218
00 7200 1800 1209600 300
```

To make a query of type **ANY** for **google.com**, use the *-a* option:

```
susel:~ # host -a google.com
Trying "google.com"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46588
;; flags: qr rd ra; QUERY: 1, ANSWER: 26, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                    IN      ANY

;; ANSWER SECTION:
google.com.             5       IN      A       208.117.229.108
google.com.             5       IN      A       208.117.229.89
google.com.             5       IN      A       208.117.229.113
google.com.             5       IN      A       208.117.229.88
google.com.             5       IN      A       208.117.229.99
google.com.             5       IN      NS      ns4.google.com.
google.com.             5       IN      MX      30 alt2.aspmx.l.google.com.
google.com.             5       IN      NS      ns3.google.com.
google.com.             5       IN      MX      10 aspmx.l.google.com.
google.com.             5       IN      A       208.117.229.98
google.com.             5       IN      SOA     ns1.google.com. dns-admin.google
.com. 1566316 7200 1800 1209600 300
google.com.             5       IN      A       208.117.229.114
google.com.             5       IN      NS      ns2.google.com.
google.com.             5       IN      MX      20 alt1.aspmx.l.google.com.
google.com.             5       IN      A       208.117.229.103
google.com.             5       IN      A       208.117.229.94
google.com.             5       IN      A       208.117.229.123
google.com.             5       IN      A       208.117.229.104
google.com.             5       IN      A       208.117.229.84
google.com.             5       IN      A       208.117.229.109
google.com.             5       IN      MX      40 alt3.aspmx.l.google.com.
```

To specify the DNS query type, use the *-t* option, followed by the type. For example, to print name servers for **google.com**, use the *-t ns* option:

```
susel:~ # host -t ns google.com
google.com name server ns2.google.com.
google.com name server ns1.google.com.
google.com name server ns3.google.com.
google.com name server ns4.google.com.
```

To print the **TXT** record (human readable information about a server) for **google.com**, use the *-t TXT* option:

```
susel:~ # host -t txt google.com
google.com descriptive text "v=spf1 include:_spf.google.com ip4:216.73.93.70/31
ip4:216.73.93.72/31 ~all"
```

Basic DNS Tool: 'dig'

Dig stands for (**Domain Information Groper**) is a network administration command-line tool for querying DNS nameservers for information about host addresses, mail exchanges, nameservers, and related information.

It is useful for verifying and troubleshooting **DNS** problems and also to perform **DNS** lookups and displays the answers that are returned from the name server that were queried. dig is part of the BIND domain name server software suite. dig command replaces older tool such as **nslookup** and the host. dig tool is available in major Linux distributions.

Install Dig on Linux

Dig is a part of DNS utility package that often gets installed with BIND name servers. You can also install the utility package that contains dig separately by accessing your VPS through **SSH** and using the following commands in the command line:

```
#dnf install bind-utils
```

Once installed, check the version, to make sure the setup was completed successfully:

```
dig -v
```

Dig Syntax

In its simplest form, the syntax of the dig utility will look like this:

```
dig [server] [name] [type]
```

[server] – the IP address or hostname of the name server to query.

If the server argument is the hostname then dig will resolve the hostname before proceeding with querying the name server.

It is optional and if you don't provide a server argument then dig uses the name server listed in **/etc/resolv.conf**.

[name] – the name of the resource record that is to be looked up.

[type] – the type of query requested by dig. For example, it can be an A record, MX record, SOA record or any other types. By default dig performs a lookup for an A record if no type argument is specified.

How to Use the Dig Command

Lets get into the basic uses of the command:

Dig a Domain Name

To perform a DNS lookup for a domain name, just pass the name along with the dig command:

```
dig hostinger.com
```

By default, the dig command will display the A record when no other options are specified. The output will also contain other information like the installed dig version, technical details about the answers, statistics about the query, a question section along with few other ones.

Short Answers

The above dig command includes a lot of useful information in different sections, but there may be times when you want only the result of the query. You can do that by using the +short option, that will display the IP address (A record) of the domain name only:

```
dig hostinger.com +short
```

Detailed Answers

Sometimes you want to view the answers section in details. Therefore, For a detailed information on answers section, you can stop displaying all the section using **+noall** option and query the answers section only by using **+answer** option with the dig command.

```
dig hostinger.com +noall +answer
```

Specifying Nameservers

By default, dig commands will query the name servers listed in **/etc/resolv.conf** to perform a DNS lookup for you. You can change this default behavior by using the **@** symbol followed by a hostname or IP address of the name server along.

The following dig command sends the DNS query to Google's name server(8.8.8.8) by using the **@8.8.8.8** option.

```
dig @8.8.8.8 atcomputer.net
```

3. Querying MX Record for Domain

Querying different types of DNS resource records only.

```
# dig atcomputer.net MX
```

4. Querying SOA Record for Domain

```
# dig atcomputer.net SOA
```

5. Querying TTL Record for Domain

```
# dig atcomputer.net TTL
```

6. Querying only answer section

```
# dig atcomputer.net +nocomments +noquestion +noauthority
+noadditional +nostats
```

7. Querying ALL DNS Records Types

```
# dig atcomputer.net ANY +noall +answer
```

8. DNS Reverse Look-up

Querying **DNS** Reverse Look-up. Only display answer section with using **+short**.

```
# dig -x 4.123.168.192 +short
```

9. Querying Multiple DNS Records

Query multiple website's DNS specific query viz. **MX**, **NS** etc. records.

```
# dig atcomputer.net mx +noall +answer redhat.com ns +noall
+answer
```