Signup Login

@bmj0114 updated at 2017-10-19

Configure DNS Wildcard with Dnsmasq Service

Network dnsmasq dns

⚠ More than 1 year has passed since last update.

This post just simply describes how to create a wildcard DNS record e.g. `*.wildcard.test.com` using dnsmasq.

> Dnsmasq: https://en.wikipedia.org/wiki/Dnsmasq
> Dnsmasq has low requirements for system resources,[6][7] can run on Linux, BSDs, Android and OS X, and is included in most Linux distributions. Consequently it "is present in a lot of home routers and certain Internet of Things gadgets"[4] and is included in Android.[5]

1. Install dnsmasq

First of all, you need to install dnsmasq service on a server which will be used as your DNS server

```
#  yum -y install dnsmasq
```

After dnsmasq is successfully installed, start and enable the service.

```
# systemctl start dnsmasq
# systemctl enable dnsmasq
```

2. Add DNS Recode

By default, dnsmasq service read `/etc/hosts` to resolve a hostname. Therefore, in order to add records to your DNS server running dnsmasq, you just need to add records `/etc/hosts` in the DNS server as below.

```
# cat /etc/hosts
127.0.0.1   localhost
10.10.10.10   dnstest.com
```

After that, restart dnsmasq service.

```
# systemctl restart dnsmasq
```

Then, you can resolve the name `dnstest.com` with the DNS server.
Login to another server and test the following command to check if the name can be resolved as expected.

```
# nslookup dnstest.com [your dns IP address]
Server:         [your dns IP address]
Address:        [your dns IP address]#53

Name:   kimitest.com
Address: dnstest.com
```

3. Test Wildcard DNS Recode (Incorrect Configuration)

No matter what you are doing, the easier the better. So I just add the following line to add wildcard record to `/etc/hosts`. (But it didn't work.)

```
# cat /etc/hosts
127.0.0.1   localhost
10.10.10.10   dnstest.com
111.111.111.111  *.wildcardtest.com
```

However, `nslookup` returned the result below.

```
# nslookup test.wildcardtest.com [your dns IP address]
Server:         [your dns IP address]
Address:        [your dns IP address]#53

** server can't find test.wildcardtest.com: NXDOMAIN
```

On the otherhand, the following command worked. It was not what I expected and completely meaningless.

```
# nslookup *.wildcardtest.com [your dns IP address]
Server:         [your dns IP address]
Address:        [your dns IP address]#53

Name:   *.wildcardtest.com
Address: 111.111.111.111
```

4. Add Wildcard DNS Recode Properly

So, how should wildcard records be added to a dnsmasq server properly?
Let's say you want both `test1.wildcardtest.com` and `test2.wildcardtest.com`, or whatever hostname with the domain `wildcardtest.com`, to be resolved to `100.100.100.100`.
It is very simple. Just add the following line to your `/etc/dnsmasq.conf`.

```
address=/wildcardtest.com/100.100.100.100
```

Or, you can also add the same line to a file under the directory `/etc/dnsmasq.d/` like below. Either way works.
As long as the file is put under the directory, you can set no matter what name to the conf-file. Automatically dnsmasq reads the files under the directory and set the configurations to its service.

```
# cat /etc/dnsmasq.d/wild-local
address=/wildcardtest.com/100.100.100.100
```

By setting the wildcard record, `*.wildcardtest.com` is going to be resolve to `100.100.100.100`.
Here is the result of testing the wildcard record.

```
# nslookup test1.wildcardtest.com [your dns IP address]
Server:         [your dns IP address]
Address:        [your dns IP address]#53

Name:   test1.wildcardtest.com
Address: 100.100.100.100

# nslookup test2.wildcardtest.com [your dns IP address]
Server:         [your dns IP address]
Address:        [your dns IP address]#53

Name:   test2.wildcardtest.com
Address: 100.100.100.100
```

```
# nslookup test3.wildcardtest.com [your dns IP address]
Server:         [your dns IP address]
Address:        [your dns IP address]#53

Name:   test3.wildcardtest.com
Address: 100.100.100.100
```

As you can see, all hostnames which has `wildcardtest.com` as its domain are resolved to `100.100.100.100` .

5. Forwarding Queries to Upstream DNS

By default, dnsmasq forwards all requests which are not able to be resolved in `/etc/hosts` to the default DNS server on the server dnsmasq is running. Therefore, you can see upstream DNS servers in `/etc/resolve.conf` like below (or maybe you need to add configuration by yourself).

```
# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver [upstream dns IP]
```

Edit request Stock LGTM 2

@bmj0114
My study logs
Follow

Why not register and get more from Qiita?