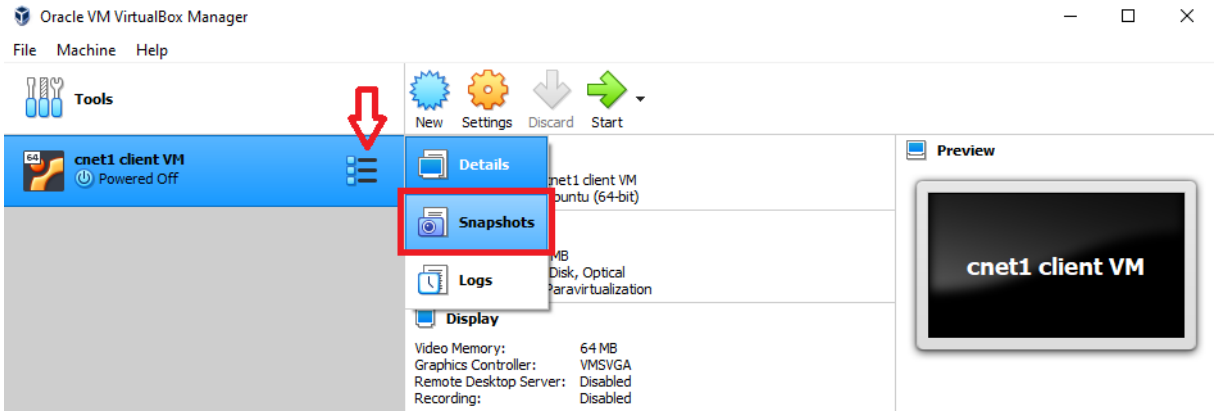
Labo computernetwerken I – DNS

DNS is een complex gedistribueerd systeem, waar miljarden clients antwoorden krijgen van talloze servers, We herhalen DNS eerst vanuit het standpunt van een client (nog niet als server).
Nadien zetten we een lokale DNS-server op, zodat we in een eigen domein ook resolving kunnen toepassen (en niet enkel op basis van IP-adressen moeten werken).

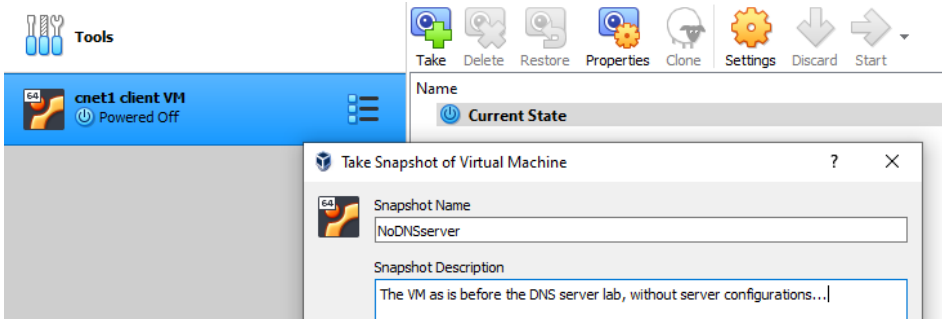
Voorbereiding: VirtualBox – snapshot[1]

Tijdens dit labo maken we wijzigingen aan de VM, die we in latere labo's niet meer nodig hebben. Meer nog: we willen nadien de wijzigingen ongedaan maken, en verder werken op de status van de VM zoals hij is in de start van dit labo. Dit kunnen we doen door middel van een *snapshot*. Een snapshot is een momentopname van de VM. Je kan op eender welk moment terugkeren naar die momentopname.

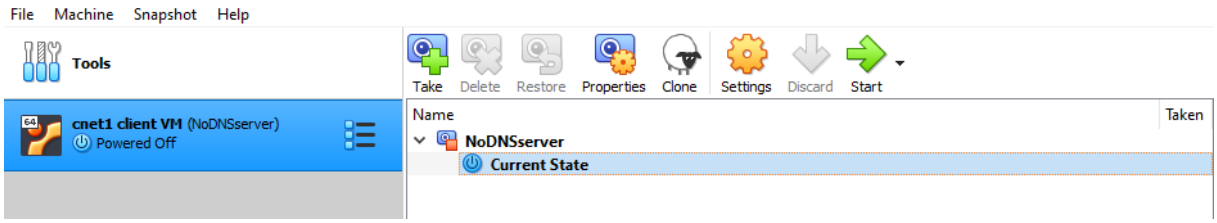
Op VirtualBox bereik je de snapshots door te klikken op de drie strepen naast je VM. We gaan ervan uit dat de VM op dit moment niet opgestart is.



Als je vervolgens op 'Take' klikt, kan je een momentopname vastpinnen. Typisch geef je een naam aan die snapshot, optioneel een omschrijving:



Hieronder vind je het verwachtte resultaat:



Hierna kan je starten met het labo DNS server.

[1] Meer uitleg op e.g.: https://www.pugetsystems.com/support/guides/virtualbox-4-using-snapshots-1914/
of ook https://docs.oracle.com/en/virtualization/virtualbox/7.1/user/Introduction.html#snapshots

Herhaling: DNS queries (vanop je client VM)

1) Query tools voor DNS clients

Een DNS client wordt door bijna elke applicatie gebruikt: een achterliggend programma vraagt aan een DNS-server op welk IP-adres correspondeert met de URL die werd opgegeven in het programma. Dit proces heet men *resolving*. Hoewel dit bijna overal achterliggend gebeurt, vindt deze *resolving* plaats bij quasi elke aanvraag die de computer start naar de buitenwereld – je merkt het gewoon niet…
Ook manueel kan een DNS request starten met een programma als **`host`** of **`nslookup`**[2]:

```
student@cnet:~$ host www.ugent.be
www.ugent.be has address 157.193.43.50
student@cnet:~$ nslookup www.ugent.be
Server:         10.0.2.1
Address:        10.0.2.1#53
Non-authoritative answer:
Name:   www.ugent.be
Address: 157.193.43.50
```

Bij het resolven wordt de default DNS-server gebruikt, die ingesteld is op de host (e.g. door DHCP). Bij **`nslookup`** worden de naam en het IP-adres van de DNS-server die ons deze informatie aanlevert eerst weergegeven, gevolgd door de naam en het IP-adres van de URL die werd opgevraagd. Behalve het nslookup commando, dat zowel werkt op Linux als op Windows, kan men op Linux ook het **`host`** commando gebruiken (kortere output), of het **`dig`** commando (extensieve output).

Wie expliciet een andere server wil gebruiken om informatie op te vragen, kan dit door bij het commando ook het server adres (of naam) mee te geven:

```
student@cnet:~$ host <gezochte URL> [DNS-server IP-adres of naam]
student@cnet:~$ host www.ugent.be 84.200.69.80
Using domain server:
Name: 8.8.4.4
Address: 8.8.4.4#53
Aliases:
www.ugent.be has address 157.193.43.50
```

Ook **`dig`** laat toe om een andere server te kiezen, en e.g. enkel een AAAA record op te vragen:

```
student@cnet:~$ dig @8.8.8.8 +short AAAA www.belnet.be
2a00:1c98:10:2c::10
```

Met **`dig`** kan je ook extra informatie opvragen, zoals bvb. de (name)servers die verantwoordelijk zijn voor de details van een bepaald domein (a.k.a. de authoritative server):

```
student@cnet:~$ dig +short NS ugent.be
ugdns1.ugent.be.
ugdns2.ugent.be.
ns.belnet.be.
```

Een *reverse lookup* – het omzetten van een IP-adres naar zijn DNS-naam – kan met **`dig`** met de optie **`-x`**:

```
student@cnet:~$ dig +short -x 157.193.215.171
www.test.atlantis.ugent.be.
```

2) DNS testen d.m.v. de tools

1. Krijg je een antwoord als je www.ugent.be resolved? Van welke DNS-server komt het antwoord?
2. Krijg je een antwoord als je www.ugent.be resolved bij server 8.8.4.4? Of bij 1.1.1.1?
3. Kan je rechtstreeks een antwoord opvragen bij de DNS-server van jouw host system?

[2] Bij Windows is dit aanwezig op de CMD prompt; installatie op Linux: **`sudo apt install dnsutils`**

DNS-server – caching

Een **caching/recursing DNS-server** is de server die door iedereen in een LAN als "zijn DNS-server" gezien wordt. Jouw laptop of desktop stelt een recursieve vraag aan deze server, die op zijn beurt iteratief het DNS-systeem raadpleegt, en de resultaten gedurende een bepaalde tijd bijhoudt voor het geval er een identieke vraag komt. Deze server wordt aangeboden door elke ISP[3], maar wordt ook geïnstalleerd in quasi elk bedrijfsnetwerk.

1) Bind default: caching server via Root DNS-servers

De DNS-(server)software op (Debian) Linux installeer je als volgt (als **root**):

```
root@cnet:~# apt install bind9
```

Deze DNS-server werkt nu enkel en alleen als server die requests doorstuurt naar het wereldwijde DNS systeem, gebaseerd op de root DNS-servers[4]. De default bind9 config werkt met beveiliging (DNSsec), dit opzetten vergt te veel tijd. We maken een leeg **options** bestand aan (zonder DNSsec), dat we tijdens dit labo verder gaan opbouwen:

```
root@cnet:~# cd /etc/bind
root@cnet:/etc/bind# mv named.conf.options named.conf.options.orig
```

Maak vervolgens een nieuw **/etc/bind/named.conf.options** aan, met enkel deze inhoud:

```
options {
 directory "/var/cache/bind";
 auth-nxdomain no;    # conform to RFC1035
};
```

Als je de configuratiebestanden hebt aangepast zal je de bind9 DNS-server software moeten herstarten om de nieuwe configuratie te activeren. Als **root** voer je de volgende commando's uit:

```
root@cnet:~# systemctl reload bind9
root@cnet:~# systemctl status bind9
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled)
[…]
Feb 28 12:15:30 cnet systemd[1]: Reloaded BIND Domain Name Server.
Feb 28 12:15:30 cnet rndc[2333]: server reload successful
Feb 28 12:15:30 cnet named[2317]: all zones loaded
Feb 28 12:15:30 cnet named[2317]: running
```

Fouten die optreden bij het uitvoeren van deze reload kunnen wijzen op fouten in de aangepaste files. Fouten worden in de log-file op locatie **/var/log/syslog** weggeschreven; de status geeft de laatste gelogde boodschappen betreffende DNS ook mooi weer.

2) Caching server met forwarder naar externe DNS-server

In het bestand **/etc/bind/named.conf.options** kan je een forwarder instellen: een externe DNS-server aan wie je zelf als server recursieve vragen stelt – zodat je als caching server niet zelf al het iteratief werk naar de root servers moet doen. Het wordt een aaneenschakeling van caching servers! Hiervoor kan je het IP van de ISP DNS-server gebruiken, of een publieke server zoals 8.8.4.4 of 1.1.1.1[5]
Voeg volgende informatie als volgt toe aan de **options** sectie (en herlaad de config in de daemon):

```
forwarders {
  <IP-adres van een externe DNS-server (en niet je eigen adres)>;
};
```

Denkvraag: kan je een DNS server die je niet kon bereiken in de vorige paragraaf, instellen als forwarder?

[3] E.g. 195.238.2.22 bij Proximus, of 195.130.130.2 bij Telenet
[4] Meer info: https://www.iana.org/domains/root/servers
[5] De meest gebruikte publieke DNS servers: https://www.wizcase.com/blog/best-free-public-dns-servers/

3) Beheer van de DNS-cache

In beide voorgaande gevallen, of je nu met een externe forwarder werkt of niet, zal jouw caching server de antwoorden die hij ontvangen heeft opslaan in zijn eigen lokale *DNS-cache*. Als je een tweede keer een identieke vraagt stelt aan een caching server (binnen een beperkte tijdsperiode), stuurt hij de vraag *niet* door naar de root servers of forwarder server, maar geeft hij het antwoord weer uit zijn cache.

`rndc` laat toe om bind, en dus ook de cache hiervan, te bewerken. Cache raadplegen kan in twee stappen:

1. Maak een dump van de cache zoals hij nu is, en sla dit op in een bestand; de default locatie en bestandsnaam is **`/var/cache/bind/named_dump.db`**

```
root@cnet:~# rndc dumpdb -cache
```

2. Open dit bestand, en ga op zoek naar de sectie van het domein waarover je de cache informatie wil bekijken. E.g. na het opvragen van www.meemoo.be vinden we dit terug in het bestand:

```
; authauthority
meemoo.be.              86386   NS      ns1.viaa.be.
                        86386   NS      ns1.belnet.be.
; authanswer
                        28786   A       185.3.217.130
; authanswer
                        28786   AAAA    2a02:5b40:4:228::7f
; authanswer
www.meemoo.be.          106     CNAME   meemoo.be.
```

De cache wissen kan door middel van het argument **`flush`**:

```
root@cnet:~# ls -l /var/cache/bind/named_dump.db
-rw-r--r-- 1 bind bind 10984 Oct 12 15:00 /var/cache/bind/named_dump.db
root@cnet:~# rndc flush
root@cnet:~# rndc dumpdb -cache
root@cnet:~# ls -l /var/cache/bind/named_dump.db
-rw-r--r-- 1 bind bind 583 Oct 12 15:01 /var/cache/bind/named_dump.db
```

De grootte van het bestand is van **`10984B`** teruggezet naar een kleine, initiële grootte van **`583B`**.

4) Opdracht: DNS caching server

1. Test met je Linux client je eigen DNS-server: vraag een query aan via jouw eigen server. Gebruik hiervoor het **`localhost`** adres.
2. Caching server **zonder** forwarder: voer de volgende zaken na elkaar uit:
   a. maak de DNS cache leeg
   b. bekijk je de DNS pakketten die langs de NIC van de server gaan met **`tcpdump`**[6]:
      ```
      root@cnet:~# tcpdump -ni eth0 udp port 53
      ```
   c. voer een query uit naar een URL. Voer hem nogmaals uit, en merk dat het antwoord komt zonder extra DNS verkeer in **`tcpdump`**!

   Werkt jouw DNS-server momenteel iteratief of recursief? Leg uit aan de hand van het verkeer dat je kunnen "capturen" hebt.
3. Werk je DNS-server bij, zodat hij een forwarder contacteert voor zijn eigen aanvragen. Herstart de bind9 server; clear de cache; monitor opnieuw het verkeer zoals in de vorige vraag. Werkt de server nu iteratief of recursief? Leg uit aan de hand het DNS-verkeer dat je kon 'buitmaken'.
4. [Extra] Een recursive server kan ook een heleboel requests ontvangen van externe hosts, extra load die je misschien liever niet wil op jouw server. Kan je beperken wie er allemaal request kan sturen naar jouw server, e.g. enkel op het localhost adres?
   Zie https://www.zytrax.com/books/dns/ch7/queries.html#allow-query

[6] **`tcpdump`** is een CLI versie van WireShark, indien nodig te installeren met **`apt install tcpdump`**

DNS-server – authoritative

Op dit moment wordt de server gebruikt door de client, maar heeft onze server geen *eigen* informatie die in de databases van bind9 opgeslagen wordt. In dit laatste deel stellen we onze eigen server in als **authoritative server**[7], weliswaar enkel voor ons eigen lokale netwerk (bestaande uit één host).

Bemerk: in een bedrijfsomgeving zal de authoritative server steeds gescheiden[8] zijn van de caching server, en bovendien ook ontdubbeld worden in een master/slave set-up. Dit valt buiten de scope van dit labo.

1) Authoritative DNS info

Per zone waarvoor de DNS-server verantwoordelijk is wordt typisch een database file gemaakt, die wordt opgenomen in de configuratiefile **`named.conf.local`**. Hieronder volgt een voorbeeld voor een DNS-server **`ns1.example.com`**:

```
zone "example.com" {
  type master;
  notify no;
  file "/etc/bind/db.example.com";
};
```

Deze regels uit **`named.conf.local`** zorgen ervoor dat de DNS-server verantwoordelijk wordt voor de zone **`example.com`**. Bovendien kan je zien dat de informatie voor deze zone in een database bestand **`db.example.com`** (in de map **`/etc/bind`**) op dezelfde machine terug te vinden is. Het uitschakelen van notificaties wordt aangeraden: dit is een (default) feature voor master/slave die we niet nodig hebben.

In dit database-bestand wordt de nodige informatie over de zone zelf bewaard:

```
$TTL    86400           ; 24 hours could have been written as 24h or 1d
example.com.  1D  IN  SOA ns1.example.com. hostmaster.example.com. (
                2002022401 ; Serial
                3H         ; Refresh
                15         ; Retry
                1w         ; Expire
                3h         ; Default TTL
                )
       IN  NS     ns1.example.com.   ; in the domain
       IN  NS     ns2.smokeyjoe.com. ; external to domain
       IN  MX  10 mail.another.com.  ; external mail provider

; server host definitions
ns1    IN  A      192.168.0.1        ; name server definition
serv1  IN  A      192.168.0.2        ; general server definition
ftp    IN  CNAME  serv1.example.com. ; ftp server definition
www    IN  CNAME  serv1              ; web server definition


; non server domain hosts
bill   IN  A      192.168.0.3
fred   IN  A      192.168.0.4
```

Merk op dat:

- de nameserver ook een A record bevat voor zichzelf.
- sommige hostnamen eindigen met een punt en andere niet. Alle namen die NIET eindigen op een punt worden geïnterpreteerd relatief t.o.v. het domein dat door de database file wordt gedefinieerd. Dit betekent dat de regel met **`www`** verwijst naar **`serv1`**, wat slaat op een machine die eigenlijk **`serv1.example.com`** heet. Controleer dit altijd goed, want dit is waarschijnlijk de meest voorkomende oorzaak van fouten.

Het opstellen van een *reverse zone* (omzetten van IP-adressen in een naam) laten we buiten beschouwing.

[7] Mochten we onze .be domeinnaam registeren, zal de TLD-server van België de domeinnaam doorverwijzen naar het IP-adres van onze server. Hierdoor wordt onze info wereldwijd opgenomen in het DNS-systeem.
[8] Zie e.g. https://kb.isc.org/docs/bind-best-practices-authoritative

2) Opdracht: Authoritative DNS

Start een lokale database op, die als domeinnaam "**<jouw familienaam>.be**" gebruikt. Voor een forward zone: voeg in de database (minstens) volgende informatie toe:

- de naamserver
- een aparte naam voor een www server (zoals fiorano.belnet.be), gebaseerd op jouw voornaam; als IP-adres geef je deze (niet-bestaande) server **10.0.2.33**
- een CNAME record die www laat verwijzen naar deze (niet-bestaande) server
- een A record voor de hostnaam van de client (e.g. jouw tweede voornaam)

Test of je alle ingestelde namen kan bereiken van je client. Je verifieert de correcte werking van je server door vanuit je client

- een NS lookup uit te voeren van de name server van jouw domein
- een lookup uit te voeren van **www.<jouw familienaam>.be**
- een lookup uit te voeren van de ingestelde client hostnaam

Vanzelfsprekend moet je jouw DNS-server (localhost) steeds meegeven met het testcommando!

DNS client – bijwerken default server

1) Client default DNS configuratie[9]

Momenteel gebruikt de host een externe DNS-server (ingesteld via de VirtualBox DHCP-server). We gaan zijn instelling veranderen zodat hij default je eigen DNS-server gebruikt om vragen aan het DNS-systeem voor hem te stellen.

De DNS-server die een client gebruikt kan je terugvinden in het bestand **/etc/resolv.conf**, e.g.:

```
search test.atlantis.ugent.be
nameserver 157.193.215.2
```

Hierboven naast het IP-adres van de nameserver, ook een toevoeging die zorgt voor het aftoetsen van korte namen binnen het domain test.atlantis.ugent.be. "**host home**" resolved op deze manier ook, en krijgt als antwoord "**home.test.atlantis.ugent.be has address 157.193.215.170**".

Als je dit manueel aanpast (wat kan), heb je een nadeel: deze informatie wordt immers aangemaakt als een interface actief wordt gebracht. Een restart, of een **ifdown/ifup** later, is deze **/etc/resolv.conf** terug overschreven. Een stuk software genaamd **resolvconf** organiseert dit. Je kan beter de basisconfiguratie van **resolvconf** bijwerken, zodat je ook bij een update van de DNS client informatie steeds opnieuw dezelfde waarde installeert in het operationele **/etc/resolv.conf**.

Een vaste waarde kan je toevoegen in het bestand **/etc/resolvconf/resolv.conf.d/base**; typisch vermeld je hierin de nameserver. Eenmaal aangepast laat je resolvconf updaten. Dit gebeurt ook automatisch bij elke reboot.

```
root@cnet:~# grep name /etc/resolv.conf
nameserver 10.0.2.1
root@cnet:~# vi /etc/resolvconf/resolv.conf.d/base # -> set to 8.8.4.4
root@cnet:~# resolvconf -u
root@cnet:~# grep name /etc/resolv.conf
nameserver 8.8.4.4
```

Bemerk dat het gebruik van de NetworkManager bij deze Linux uitgeschakeld is – zie de voetnoot.

2) Opdracht: default DNS bijwerken

Pas je Linux client aan: zorg dat je eigen DNS-server (**127.0.0.1**) altijd de enige DNS-server zal zijn voor deze client.

[9] Voor een volledig detail over netwerk- en DNS-instellingen: https://wiki.debian.org/NetworkConfiguration

Extra: hosts file

De **hosts** file is een bestand die lokaal op je computer aanwezig is, waarin je hostnamen aan IP-adressen kan koppelen. Op Linux vind je dit bestand in de map **/etc**[10]. De inhoud doet ons ook inzien waarom het woord '**localhost**' effectief werkt op de CLI!

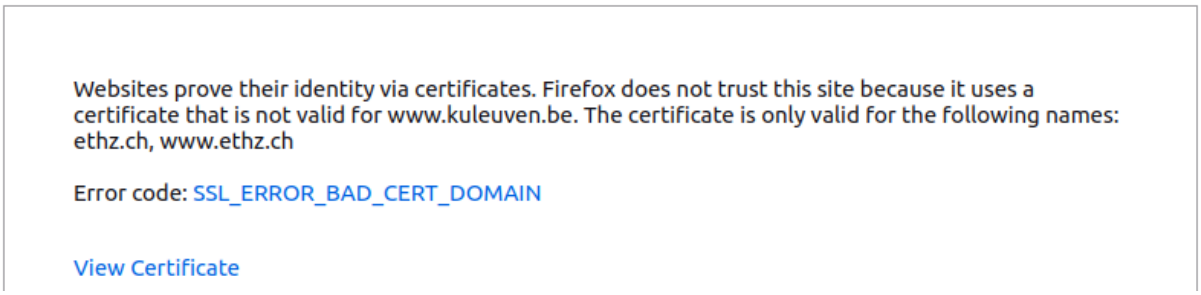
```
root@cnet:~# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       cnet.vm       cnet

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

De informatie in dit bestand krijgt voorrang op het DNS-systeem. Kortom: bij elke vraag wordt er eerst nagegaan of de URL zich niet in de **hosts** file bevindt. E.g. als er een handmatige entry voor een website wordt toegevoegd in dit bestand voor één van de servers uit het eerste labo, zal de DNS info niet langer opgevraagd worden voor doorsnee programma's:

```
root@cnet:~# grep kul /etc/hosts
129.132.19.216  www.kuleuven.be
root@cnet:~# ping www.kuleuven.be
PING www.kuleuven.be (129.132.19.216) 56(84) bytes of data.
64 bytes from www.kuleuven.be (129.132.19.216): icmp_seq=1 ttl=46 time=23.8 ms
```

Bemerk: een programma als **host** of **dig** raadpleegt steeds rechtstreeks de DNS-systemen, en slaat het hosts bestand over. Deze tools zijn dus niet geschikt om dit te gaan testen - ping of surfen wel.

Let wel: huidige websites en browsers werken allemaal via HTTPs, waar na de TCP 3-way handshake er ook steeds een TLS handshake is die de naam van de site aftoetst aan de hand van een certificaat. Als dit niet overeenstemt, krijg je een error zoals hieronder:



Nu je toch aan de slag bent kan je dit proberen reproduceren!

1) Opdracht: hosts file

Je kan een website die je nooit wil bezoeken opnemen in de hosts file. Hierdoor zal je altijd naar het IP-adres gaan dat je opgeeft in deze hosts file! Koppel bvb. www.flair.be aan het IP-adres 157.193.215.171 en merk dat je inderdaad niet meer kan surfen met firefox naar deze website!

[10] Dit bestaan ook op andere besturingssytemen, zie bvb. https://www.howtogeek.com/27350/beginner-geek-how-to-edit-your-hosts-file/

Cleanup: VirtualBox – snapshot (again)

Als je klaar bent met dit lab, heb je een DNS-server ter referentie. Je kan deze configuratie opslaan in een nieuwe snapshot, en op deze lijn niet meer verder werken. Nadien kan je de snapshot van het begin van het lab herstellen, en verder werken op de oorspronkelijke VM. Je hebt nu, met bijna niks van extra schijfruimte, een versie waarbinnen wel een DNS-server is opgezet, en een versie waar we de volgende labo's op verder kunnen werken.