

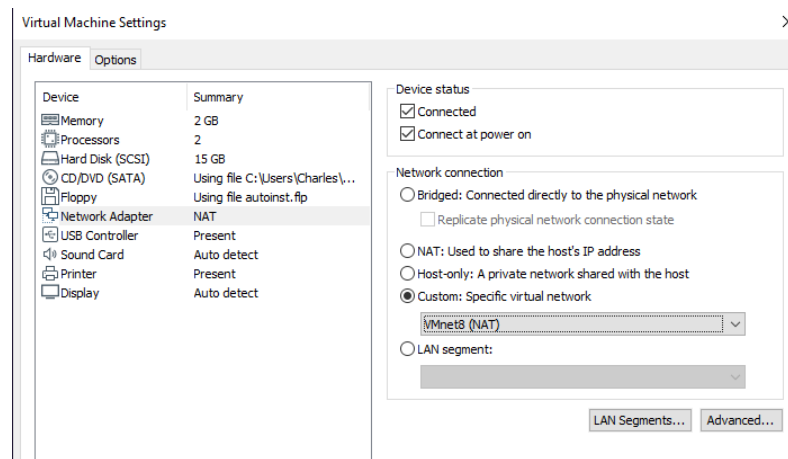
Procédures d'installation et de configuration des serveurs

Installation et configuration d'un système d'exploitation

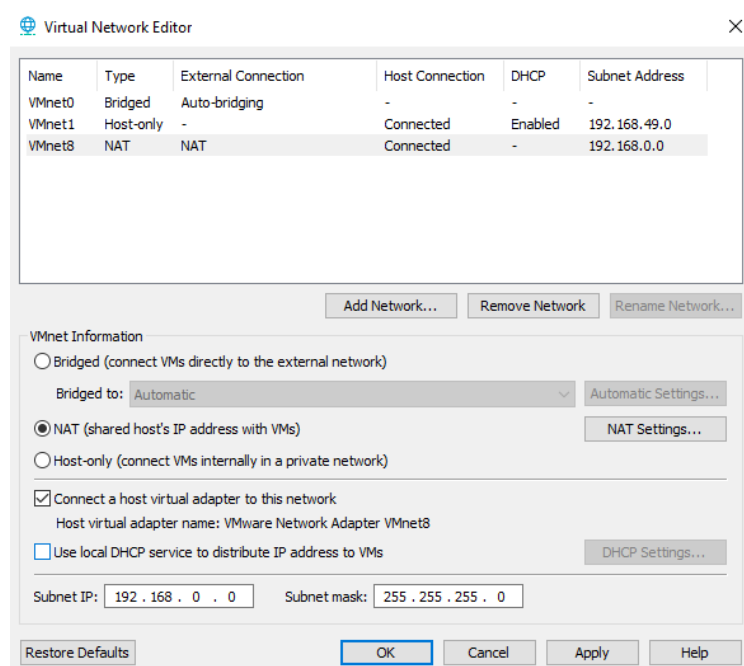
Dans un premier temps, il va falloir créer les machines sur lesquelles nous allons héberger nos serveurs et également configurer leur système d'exploitation.

Toutes les machines utilisées ici vont être des machines virtuelles Linux sous Debian9 avec une interface MATE Desktop Environment 1.16.2 basée sur gnome

Sous VMWare, nous allons configurer toutes les machines en VMnet8 (NAT) :

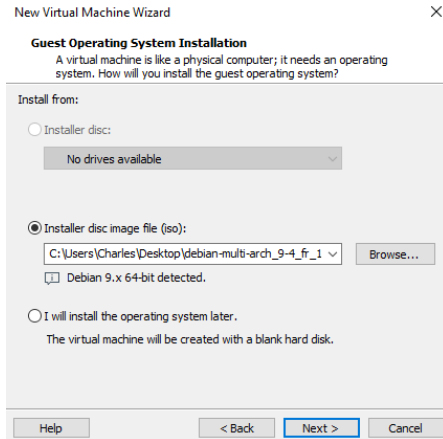


Il faudra ensuite que VMNet8 soit configuré dans le Virtual Network Editor sur le réseau 192.168.0.0/24 (dans Virtual Network Editor) et que la case « Use local DHCP service to distribute IP address to VMs » soit décochée.



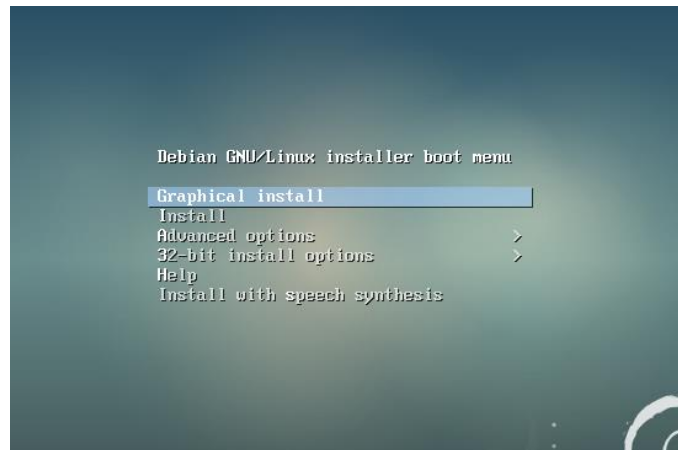
Dans un tout premier temps, il va falloir une image ISO du système d'exploitation à installer. Ici Debian9 à télécharger par exemple sur le site de Debian : <https://www.debian.org/distrib/netinst>

Une clé bootable sera à configurer si on veut installer l'OS sur une machine physique, on donnera ici l'ISO à VMWare Workstation pour commencer notre installation de Debian :

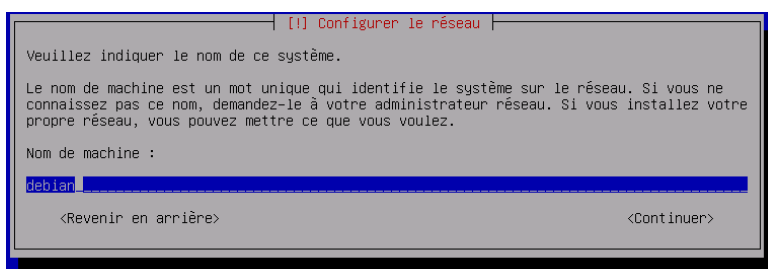


On choisira également ensuite le nom de la machine ainsi que l'espace disque à attribuer.

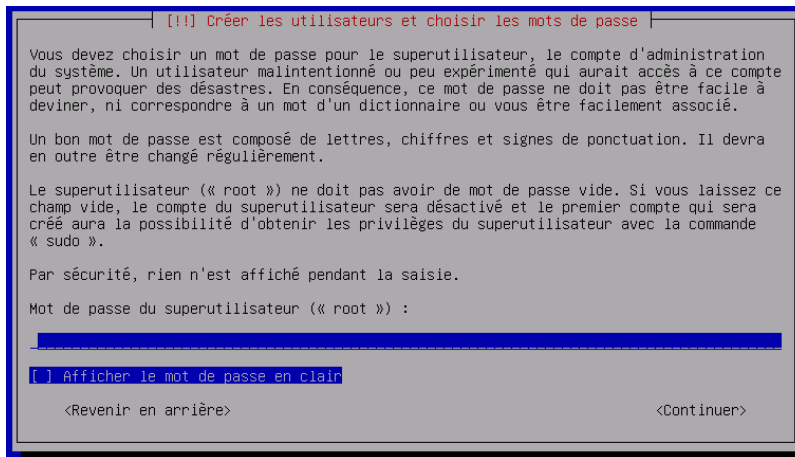
Une fois la machine démarrée, on va pouvoir commencer l'installation et la configuration de Debian.



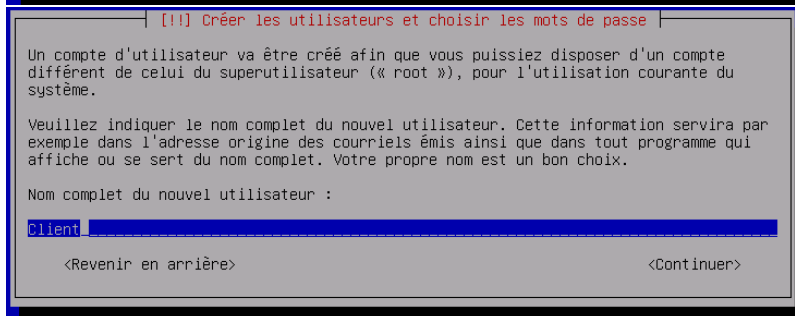
On choisira alors « graphical install » ou « install » si l'on ne veut pas une installation avec une interface graphique. On pourra ensuite sélectionner notre langue.



On entrera ensuite le nom de la machine.

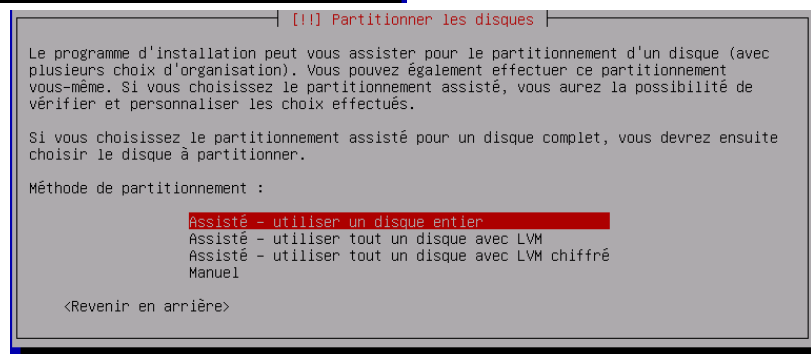


Puis on créera un mot de passe super utilisateur.



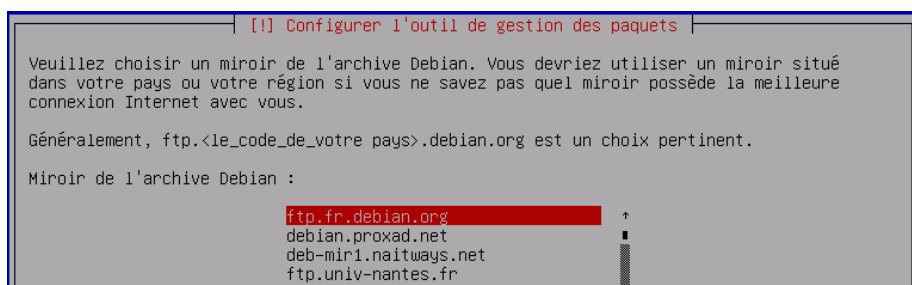
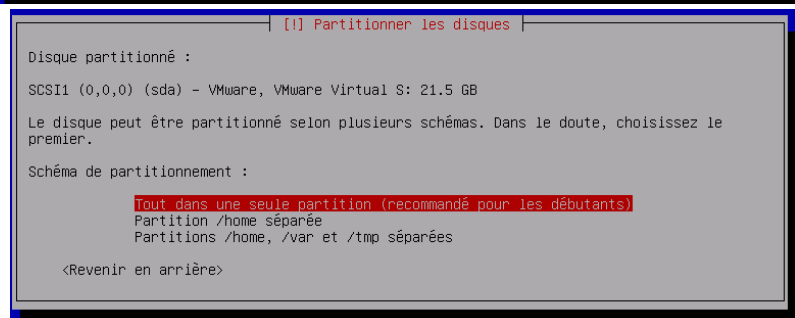
Et on créera un nouvel utilisateur puis on rentrera un mot de passe par la suite.

On utilisera un disque entier.



On sélectionnera « Tout dans une seule partition ».

Puis on validera.



On sélectionnera le premier miroir.

Le système va alors télécharger et installer les paquets qui lui sont utiles. On se retrouvera finalement avec une interface en ligne de commandes dans laquelle nous pourrons saisir une commande pour

installer une interface graphique selon nos envies. Par exemple la commande `apt-get install gnome` permettra d'installer l'interface graphique gnome sur le système (une des plus répandues).

Toutes les commandes à suivre seront principalement à rentrer en tant qu'administrateur, il est donc conseillé de passer en mode « super-utilisateur » à chaque ouverture d'une nouvelle fenêtre terminal avec la commande `su` sur chaque machine :

A screenshot of a terminal window titled 'gonfreecs@debian: ~'. The window has a menu bar with 'Fichier', 'Édition', 'Affichage', 'Rechercher', 'Terminal', and 'Aide'. The terminal shows the prompt 'gonfreecs@debian:~\$' followed by the command 'su'. It then prompts 'Mot de passe :', and after an invisible password, it shows the root prompt 'root@debian:/home/gonfreecs#' with a cursor.

Serveur DNS maître

Le serveur DNS maître va faire le lien entre le nom de domaine et l'adresse IP.

Procédure d'installation du serveur DNS maître :

Avant tout, il est conseillé d'installer les paquets qui nous seront utiles par la suite tant qu'un accès à internet est en place (il sera sinon toujours possible par la suite de retrouver internet mais cela nécessite plusieurs manipulations) :

- `apt-get update`
- `apt-get upgrade`
- `apt-get install ifupdown`
- `apt install isc-dhcp-server`
- `aptitude install bind9`
- `apt install curl`

Selon vos préférences vous pouvez installer un éditeur de fichier tel qu'emacs :

- `apt install emacs`

la commande nano pourra sinon très bien effectuer le travail.

Tout d'abord, il faut configurer la machine avec une adresse IP statique, paramétrée par défaut en DHCP :

Ouvrir le fichier `/etc/network/interfaces` (avec Emacs ou Nano) et modifier « DHCP » en « static » puis ajouter l'adresse voulue et le masque correspondant :

```
Applications Emplacements Système
gonfreecs@master: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.10.5
netmask 255.255.255.0
dns-nameservers 192.168.10.5 192.168.10.6
```

Ici on utilise l'adresse 192.168.10.5 avec le netmask 255.255.255.0.

On ajoutera, pour éviter d'y revenir, les adresses DNS de nos 2 serveurs (maître et esclave) afin que le fichier resolv.conf ne soit pas remis à zéro lors d'un prochain redémarrage. :

```
dns-nameservers 192.168.10.5 192.168.10.6
```

On relance ensuite le service network pour valider nos changements avec la commande :

- service networking restart
- ifdown eth0
- ifup eth0

(Attention à vérifier le nom de votre interface réseau, ici c'est eth0)

On peut ensuite vérifier les changements avec la commande ifconfig :

```
root@master:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.10.5  netmask 255.255.255.0  broadcast 192.168.10.255
    inet6 fe80::20c:29ff:fe8c:edc2  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:8c:ed:c2  txqueuelen 1000  (Ethernet)
    RX packets 23577  bytes 2004448 (1.9 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 22895  bytes 2229469 (2.1 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1  (Boucle locale)
    RX packets 15048  bytes 1209095 (1.1 MiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 15048  bytes 1209095 (1.1 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Une fois l'IP modifiée, il va falloir modifier les fichiers de configuration /etc/hostname et /etc/hosts

On va définir le nom FQDN de la machine (par défaut : Debian) dans le fichier /etc/hostname :

```
gonfreecs@master: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
GNU nano 2.7.4 Fichier : /etc/hostname  
master.carnofluxe.domain
```

On choisira le nom *master.carnofluxe.domain* pour le DNS Maître.

On modifie ensuite le fichier */etc/hosts* permettant une résolution locale des noms comme suit :

```
gonfreecs@master: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
GNU nano 2.7.4 Fichier : /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 master  
192.168.10.5 master.carnofluxe.domain master  
192.168.10.6 slave.carnofluxe.domain slave  
  
# The following lines are desirable for IPv6 capable hosts  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

Enfin, il va également être nécessaire de configurer le fichier */etc/resolv.conf* pour que le serveur soit intégré à la future zone DNS. Il va donc falloir indiquer dans ce fichier le domaine et la zone de recherche DNS :

```
gonfreecs@master: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
GNU nano 2.7.4 Fichier : /etc/resolv.conf  
domain carnofluxe.domain  
search carnofluxe.domain  
nameserver 192.168.10.5  
nameserver 192.168.10.6
```

On y ajoute également l'adresse de notre DNS esclave (192.168.10.6) afin d'éviter de retourner dans ce fichier lors de la configuration de ce dernier.

Configuration du serveur DNS maître :

Tous les paramètres de bases sont maintenant définis, on va alors pouvoir configurer le DNS avec bind9.

On va tout d'abord se rendre dans le répertoire */etc/bind* avec la commande *cd* afin d'accéder aux fichiers de configuration plus rapidement.

Deux zones DNS seront à déclarer : la zone *carnofluxe.domain* et la zone inverse associée *10.168.192.in-addr.arpa* (permettant une traduction des adresses IP en noms de domaine)

Ces zones se déclarent dans le fichier *named.conf.local* du répertoire */etc/bind*, pour chaque zone, il faut préciser que le serveur est maître ou esclave et le fichier contenant les informations sur la zone.

Voici la déclaration des zones à ajouter dans le fichier */etc/bind/named.conf.local* :

```
gonfreecs@master: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "carnofluxe.domain" {
    type master;
    allow-transfer {192.168.10.6; };
    notify yes;
    file "/etc/bind/db.carnofluxe.domain";
};
zone "10.168.192.in-addr.arpa" {
    type master;
    allow-transfer {192.168.10.6; };
    notify yes;
    file "/etc/bind/rev.carnofluxe.domain";
};
```

Il est ensuite possible de préciser des options de configuration dans le fichier `/etc/bind/named.conf.options`:

```
gonfreecs@master: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : named.conf.options

options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    // forwarders {
    //     0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
    //=====
    #dnssec-validation auto;

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
    listen-on { 192.168.10.5; 192.168.10.6; };
    recursion no;
    allow-query-cache { none; };
    version none;
    additional-from-cache no;
    allow-transfer { 192.168.10.6; };
};
```

On va ensuite devoir créer les fichiers de zones définis dans `named.conf.local`: ici `db.carnofluxe.domain` et `rev.carnofluxe.domain` grâce à la commande `touch`.

On configure le premier comme ceci :

```

gonfreecs@master: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : db.carnofluxe.domain

$TTL 10800
@      IN SOA master.carnofluxe.domain.      root.carnofluxe.domain. (
      2016121801;
      3h;
      1h;
      1w;
      1h);
@      IN      NS      master.carnofluxe.domain.
@      IN      NS      slave.carnofluxe.domain.

master IN A      192.168.10.5
slave  IN A      192.168.10.6
site   IN A      192.168.10.10
superv IN A      192.168.10.10

www     IN      CNAME   site

```

SOA (*Start Of Authority*) proclame des informations importantes faisant autorité sur un espace de nom pour le DNS. C'est le premier élément à indiquer dans un fichier de zone.

Structure d'un SOA :

```

@      IN      SOA      <primary-name-server>      <hostmaster-email> (
      <serial-number>
      <time-to-refresh>
      <time-to-retry>
      <time-to-expire>
      <minimum-TTL> )

```

Le symbole @ place le nom de zone en tant qu'espace de nom défini par le SOA.

Ensuite, les « NS » correspondent au nameserver faisant autorité sur la zone, le « A » est un enregistrement d'adresse qui spécifie une adresse IP à assigner à un nom. (Par exemple 192.168.168.10.6 est assigné à slave.carnofluxe.domain)

Enfin « CNAME » nous permettra d'avoir un alias en plus du nom spécifié (www.carnofluxe.domain et site.carnofluxe.domain redirigeront par exemple au même endroit)

Le second fichier sera configuré comme cela (pour la zone inverse) :

```

gonfreecs@master: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : rev.carnofluxe.domain

$TTL 10800
@      IN SOA master.carnofluxe.domain.      root.carnofluxe.domain. (
      2016121801;
      3h;
      1h;
      1w;
      1h);
@      IN      NS      master.carnofluxe.domain.
@      IN      NS      slave.carnofluxe.domain.

5      PTR      master.carnofluxe.domain.
6      PTR      slave.carnofluxe.domain.
10     PTR      site.carnofluxe.domain.
10     PTR      superv.carnofluxe.domain.

```

Les chiffres devant les PTR correspondent au dernier nombre ou chiffre de l'adresse IP correspondant au nom de domaine, « PTR » est un enregistrement de pointeur servant à la résolution inverse, il renvoie les adresses IP vers un nom de domaine. (Par exemple, l'IP 192.168.10.5 sera renvoyée vers master.carnofluxe.domain)

Les fichiers de zones maintenant créés, il ne reste plus qu'à vérifier leur configuration puis leur fonctionnement :

Avec la commande *named.checkconf*, on peut savoir si la configuration des fichiers de configuration est bien traitée, si elle ne renvoie rien c'est que la configuration est valide.

On utilise ensuite la commande *named-checkzone* « *nom de la zone* » « *nom du fichier* » pour tester la configuration des fichiers de zone :

```
root@master:/etc/bind# named-checkconf
root@master:/etc/bind#
root@master:/etc/bind# named-checkzone carnofluxe.domain db.carnofluxe.domain
zone carnofluxe.domain/IN: loaded serial 2016121801
OK
root@master:/etc/bind# named-checkzone 10.168.192.in-addr.arpa rev.carnofluxe.domain
zone 10.168.192.in-addr.arpa/IN: loaded serial 2016121801
OK
root@master:/etc/bind#
```

On remarque ici que tout semble bien configuré.

Il ne reste plus qu'à redémarrer le service bind9 avec :

```
- service bind9 restart
```

La commande `service bind9 status` nous permettra de vérifier que le service est bien démarré :

```
root@master:/etc/bind# service bind9 status
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2019-02-11 16:14:31 CET; 8h ago
     Docs: man:named(8)
  Process: 2613 ExecStop=/usr/sbin/rndc stop (code=exited, status=0/SUCCESS)
 Main PID: 2617 (named)
    Tasks: 5 (limit: 19660)
   CGroup: /system.slice/bind9.service
           └─2617 /usr/sbin/named -f -u bind

févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone 0.in-addr.arpa/IN: loaded serial 1
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone 127.in-addr.arpa/IN: loaded serial 1
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone 10.168.192.in-addr.arpa/IN: loaded serial 2016121801
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone 255.in-addr.arpa/IN: loaded serial 1
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone carnofluxe.domain/IN: loaded serial 2016121801
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone localhost/IN: loaded serial 2
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: all zones loaded
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: running
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone 10.168.192.in-addr.arpa/IN: sending notifies (serial 2016121801)
févr. 11 16:14:31 master.carnofluxe.domain named[2617]: zone carnofluxe.domain/IN: sending notifies (serial 2016121801)
root@master:/etc/bind#
```

On peut maintenant essayer une résolution de noms avec la commande *dig* « *nom de domaine* » :

```

root@master:/etc/bind# dig www.carnofluxe.domain

; <<>> DiG 9.10.3-P4-Debian <<>> www.carnofluxe.domain
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 46354
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.carnofluxe.domain.      IN      A

;; ANSWER SECTION:
www.carnofluxe.domain. 10800 IN      CNAME  site.carnofluxe.domain.
site.carnofluxe.domain. 10800 IN      A      192.168.10.10

;; AUTHORITY SECTION:
carnofluxe.domain.    10800 IN      NS      slave.carnofluxe.domain.
carnofluxe.domain.    10800 IN      NS      master.carnofluxe.domain.

;; ADDITIONAL SECTION:
slave.carnofluxe.domain. 10800 IN      A      192.168.10.6
master.carnofluxe.domain. 10800 IN      A      192.168.10.5

;; Query time: 0 msec
;; SERVER: 192.168.10.5#53(192.168.10.5)
;; WHEN: Tue Feb 12 00:36:51 CET 2019
;; MSG SIZE rcvd: 158

root@master:/etc/bind# █

```

Et vérifier qu'on obtient bien une réponse valide.

Pour essayer une résolution de noms inverse :

```

root@master:/etc/bind# dig -x 192.168.10.6

; <<>> DiG 9.10.3-P4-Debian <<>> -x 192.168.10.6
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15687
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;6.10.168.192.in-addr.arpa.  IN      PTR

;; ANSWER SECTION:
6.10.168.192.in-addr.arpa. 10800 IN      PTR      slave.carnofluxe.domain.

;; AUTHORITY SECTION:
10.168.192.in-addr.arpa. 10800 IN      NS      master.carnofluxe.domain.
10.168.192.in-addr.arpa. 10800 IN      NS      slave.carnofluxe.domain.

;; ADDITIONAL SECTION:
slave.carnofluxe.domain. 10800 IN      A      192.168.10.6
master.carnofluxe.domain. 10800 IN      A      192.168.10.5

;; Query time: 0 msec
;; SERVER: 192.168.10.5#53(192.168.10.5)
;; WHEN: Tue Feb 12 00:38:53 CET 2019
;; MSG SIZE rcvd: 158

root@master:/etc/bind# █

```

Le serveur DNS maître est donc maintenant fonctionnel.

Serveur DNS esclave

Le serveur DNS esclave va prendre le relais si le serveur DNS maître ne fonctionne plus.

Procédure d'installation du serveur DNS esclave :

Lors d'une installation de DNS, il est fortement conseillé de mettre en place une architecture maître-esclave, la configuration du second est ainsi très proche du premier :

Évidemment, il va falloir installer les paquets utiles avec les commandes données pour l'installation du DNS maître. De plus, nous allons avoir des manipulations à faire avec des scripts par la suite, il va donc falloir en plus installer ces paquets :

- apt-get install mailutils
- apt-get install mutt
- apt-get install ssh
- apt-get install openssh-server
- apt-get install sshpass

Il va ainsi falloir commencer par changer l'adresse IP du serveur esclave en un serveur DNS en suivant les mêmes instructions que ci-dessus. On lui attribuera cette fois l'IP 192.168.10.6

```
gonfreecs@slave: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 192.168.10.6
netmask 255.255.255.0
dns-nameservers 192.168.10.5 192.168.10.6
```

Pour le nom FQDN, on choisira ici *slave.carnofluxe.domain*

```
gonfreecs@slave: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : /etc/hostname

slave.carnofluxe.domain
```

Les fichiers `/etc/hosts` et `/etc/resolv.conf` seront à modifier exactement de la même façon que sur le DNS maître.

Configuration du serveur DNS esclave :

Pour la configuration du serveur DNS esclave, la manipulation est plus simple puisqu'elle s'appuie sur celle du DNS maître, on modifiera donc le fichier `/etc/bind/named.conf.local` ainsi :

```
gonfreecs@slave: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.7.4                                Fichier : named.conf.local

//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "carnofluxe.domain" {
    type slave;
    masters { 192.168.10.5; };
    file "/var/cache/bind/db.carnofluxe.domain";
};
zone "10.168.192.in-addr.arpa" {
    type slave;
    masters { 192.168.10.5; };
    file "/var/cache/bind/rev.carnofluxe.domain";
};
```

Les fichiers de zones seront ainsi générés automatiquement dans les répertoires donnés avec les informations qui seront récupérées depuis le serveur maître.

(Attention : il est important de préciser un chemin différent que `/etc/bind` pour ces fichiers de zone car il va être impossible pour le système d'écrire dans cet espace automatiquement.)

Il suffit maintenant de redémarrer le service avec la commande :

```
- service bind9 restart
```

Et vérifier son status avec `service bind9 status`. On peut également se rendre compte que les fichiers ont été créés à l'endroit spécifié :

```
root@slave:/etc/bind# cd /var/cache/bind
root@slave:/var/cache/bind# ls
db.carnofluxe.domain  managed-keys.bind  rev.carnofluxe.domain
root@slave:/var/cache/bind# █
```

Pour faire un test, on pourra utiliser la commande `dig` en spécifiant le nom de domaine est l'IP du DNS depuis lequel on veut faire la résolution de noms (ici lui-même) :

```
root@slave:/var/cache/bind# dig www.carnofluxe.domain @192.168.10.6

; <<<> DiG 9.10.3-P4-Debian <<> www.carnofluxe.domain @192.168.10.6
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 10658
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.carnofluxe.domain.      IN      A

;; ANSWER SECTION:
www.carnofluxe.domain.  10800   IN      CNAME   site.carnofluxe.domain.
site.carnofluxe.domain. 10800   IN      A       192.168.10.10

;; AUTHORITY SECTION:
carnofluxe.domain.      10800   IN      NS       slave.carnofluxe.domain.
carnofluxe.domain.      10800   IN      NS       master.carnofluxe.domain.

;; ADDITIONAL SECTION:
slave.carnofluxe.domain. 10800   IN      A       192.168.10.6
master.carnofluxe.domain. 10800   IN      A       192.168.10.5

;; Query time: 0 msec
;; SERVER: 192.168.10.6#53(192.168.10.6)
;; WHEN: Tue Feb 12 01:01:45 CET 2019
;; MSG SIZE rcvd: 158

root@slave:/var/cache/bind# █
```

Serveur HTTP

Le serveur HTTP qui sert à héberger le site que nous avons créé.

Procédure d'installation du serveur HTTP :

Pour le serveur HTTP, les paquets à installer en premier lieu sont plus nombreux que pour le DNS/DHCP :

- apt-get update
- apt-get upgrade
- apt-get install ifupdown
- apt install apache2
- apt-get install mailutils
- apt-get install mutt
- apt-get install ssh
- apt-get install openssh-server
- apt-get install sshpass
- apt install rsync
- apt-get install exfat-fuse exfat-utils

Ce dernier paquet permet l'installation d'Apache, serveur HTTP très populaire que l'on va utiliser ici.

Avec Apache, dans notre cas, chaque site web va correspondre à un hôte virtuel qui va être défini par un fichier de configuration indépendant situé dans le répertoire `/etc/apache2/sites-available/`

On remarquera qu'un hôte virtuel correspondant au fichier `000-default.conf` est déjà présent, pouvant servir d'exemple. Nous n'y toucherons pas ici et préféreront créer notre propre VirtualHost.

On va ainsi créer notre premier VirtualHost pour héberger notre site de e-commerce. Grâce à la commande `touch` on créera dans ce répertoire un fichier `site.carnofluxe.domain.conf` tel quel :

```
gonfreecs@debian: ~  
Fichier Édition Affichage Rechercher Terminal Aide  
GNU nano 2.7.4 Fichier : site.carnofluxe.domain.conf  
  
<VirtualHost *:80>  
    ServerName site.carnofluxe.domain  
    ServerAlias www.site.carnofluxe.domain  
    DocumentRoot "/var/www/html"  
    <Directory "/var/www/html">  
        Options +FollowSymLinks  
        AllowOverride all  
        Require all granted  
    </Directory>  
    ErrorLog /var/log/apache2/error.site.carnofluxe.domain.log  
    CustomLog /var/log/apache2/access.site.carnofluxe.domain.log combined  
</VirtualHost>
```

On remarquera ainsi que l'on a donné dans ce fichier un chemin vers `/var/www/` qui est le répertoire où sont stockés les fichiers HTML des serveurs.

Pour ce premier VirtualHost on donnera le chemin `/var/www/html` qui est le dossier de la page affichée par défaut lorsque l'on tape l'adresse IP du serveur dans un navigateur du réseau. Cette manipulation va ainsi nous permettre d'afficher le site de e-commerce soit avec l'IP soit avec le nom de domaine `site.carnofluxe.domain` (ou www.carnofluxe.domain).

On est ensuite capable d'activer cette configuration avec la commande

```
- a2ensite site.carnofluxe.domain
```

puis on recharge la configuration d'apache :

```
- systemctl reload apache2
```

Configuration du serveur HTTP :

On va maintenant pouvoir aller trouver le fichier `index.html` dans le dossier `/var/www/html`. On remarquera que ce fichier html contient déjà beaucoup de chose, c'est en fait la page d'accueil d'Apache2 qui est chargée normalement lorsque l'on accède au serveur avec son IP. Dans notre cas on souhaite que la page qui s'affiche avec l'adresse IP soit la page web de notre site de e-commerce.

On va donc modifier le contenu de ce fichier `index.html` comme ci :

(Je conseille de faire une sauvegarde du fichier `index.html` contenant la page apache si jamais on souhaite restaurer la page par défaut. Par exemple avec la commande :

```
- mv index.html index.html.backup )
```



```
gonfreecs@debian: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.7.4      Fichier : index.html

<!DOCTYPE html>
<html>
<head>
  <title> Carnofluxe </title>
  <meta charset="UTF-8" />
</head>
<body>
  <p> Bonjour, ceci est le site de e-commerce Carnofluxe </p>
</body>
</html>
```

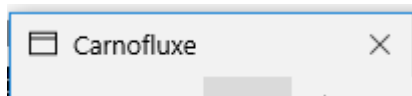
On peut remarquer les différentes parties du fichier représentées grâce aux balises html (se présentant sous la forme

`< « nom de la balise »>`

...

`< / « nom de la balise »>)`

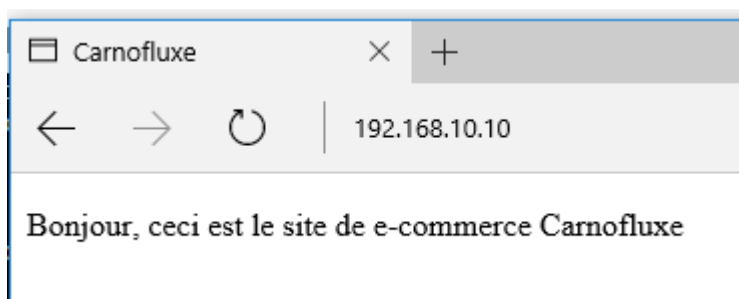
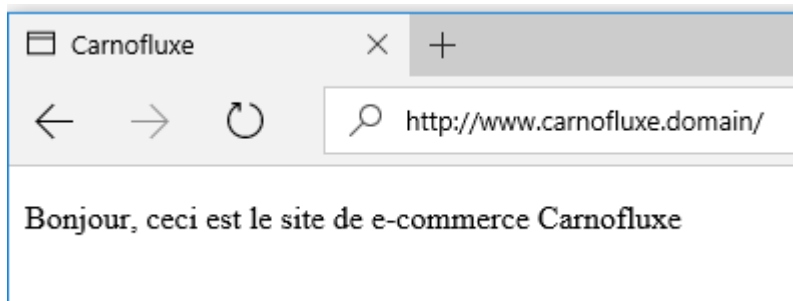
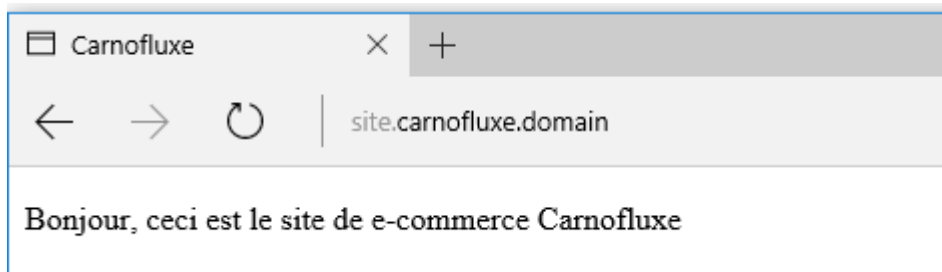
Le « head » nous permettra de modifier l'en-tête de la page html, par exemple on aura ici le titre « Carnoflux » comme ceci :



La ligne `<meta charset= 'UTF_8 ' />` nous permettra d'insérer des caractères spéciaux comme des accents dans le titre et le corps de notre page.

Le « body » nous permettra de modifier tout le corps de la page html, on peut y insérer du texte, des images, des liens, des gifs animés etc.

Notre page web est maintenant en place et fonctionnelle, on remarquera une fois les autres serveurs installés (DNS maître-esclave et DHCP) et un client dans le réseau connecté au DHCP configuré que la page web est accessible aux noms de domaines configurés dans le DNS. Ici la page sera chargée si on rentre `site.carnoflux.domain`, www.carnoflux.domain ou l'IP `192.168.10.10` dans un navigateur :



Maintenant, dans notre cadre nous avons besoin d'un site de supervision hébergé sur le même serveur avec un second hôte virtuel, on va alors suivre des manipulations similaires :

Tout d'abord créer un fichier `superv.carnoflux.domain` dans le répertoire `/etc/apache2/sites-available/` :

```
gonfreecs@debian: ~
Fichier Édition Affichage Rechercher Terminal Aide
GNU nano 2.7.4 Fichier : superv.carnofluxe.domain.conf

<VirtualHost *:80>
    ServerName superv.carnofluxe.domain
    ServerAlias www.superv.carnofluxe.domain
    DocumentRoot "/var/www/superv.carnofluxe"
    <Directory "/var/www/superv.carnofluxe">
        Options +FollowSymLinks
        AllowOverride all
        Require all granted
    </Directory>
    ErrorLog /var/log/apache2/error.superv.carnofluxe.domain.log
    CustomLog /var/log/apache2/access.superv.carnofluxe.domain.log combined
</VirtualHost>
```

On lui donne un chemin vers un nouveau dossier dans `/var/www/` dans lequel on pourra configurer l'`index.html`

(On n'oubliera pas d'activer la configuration avec les commandes :

- `a2ensite site.carnofluxe.domain`
- `systemctl reload apache2`)

Pour la suite, notre fichier `/var/www/superv.carnofluxe/index.html` sera généré automatiquement par un script qui récupère un fichier csv généré sur le serveur DNS esclave et le transforme en page html. (Voir la documentation sur les scripts)

Exemple de script généré :

```
Ouvrir  index.html [Lecture seule]  Enregistrer
/var/www/superv.carnofluxe

<html>
  <head>
    <meta charset=utf-8>
    <title>Adresses clients</title>
  </head>
  <body>
    <h1>Nombre de clients s'étant connectés durant la dernière heure : 1<br/>
>192.168.10.6,</h1>
    <body>
    <h1>Connexion établie entre votre machine et le serveur HTTP !<br>
Adresse ip du site : 192.168.10.10 </br>
Temps de connexion avec le site : 0.337 ms </br>
Temps de chargement de la page : 0s </br></h1>
    </body>
  </html>
```

Serveur DHCP

Le serveur DHCP attribuera automatiquement une adresse IP (d'une plage d'adresses donnée) à un ordinateur configuré en DHCP dans le réseau local.

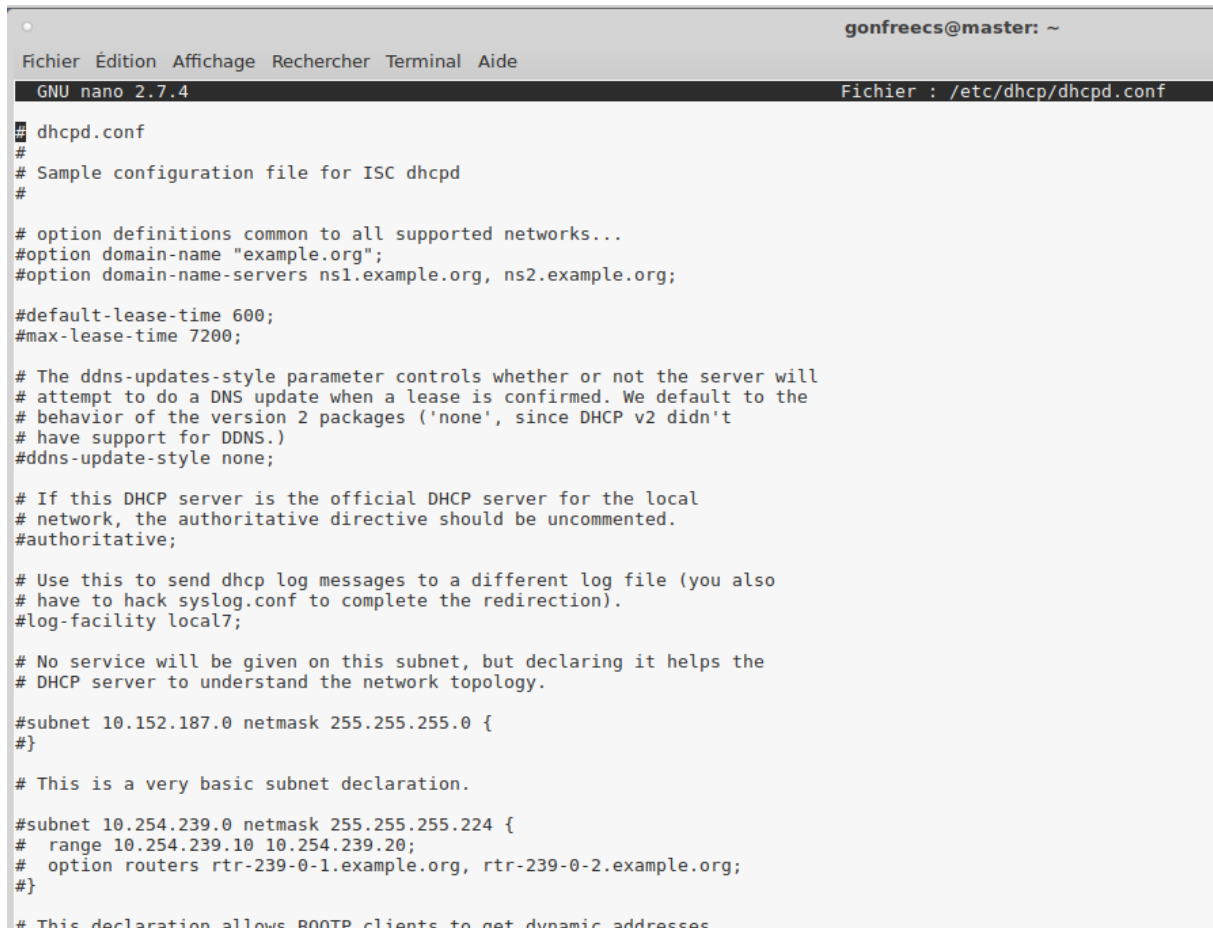
Installation du serveur DHCP

Pour notre installation DHCP, nous utilisons le même serveur que pour le DNS maître. Les paquets utiles ont donc normalement déjà été installés et la machine est normalement déjà configurée avec une adresse IP statique (192.168.10.5).

Configuration du serveur DHCP

Il va maintenant falloir aller dans le fichier de configuration DHCP pour configurer ce dernier. Ils se trouve dans le répertoire `/etc/dhcp/dhcpd.conf` (l'ouvrir dans l'invite de commande en root avec nano).

Le fichier se présente comme ceci :



```
gonfreecs@master: ~
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.7.4                               Fichier : /etc/dhcp/dhcpd.conf

# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
#default-lease-time 600;
#max-lease-time 7200;
# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
#ddns-update-style none;
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;
# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;
# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.
#subnet 10.152.187.0 netmask 255.255.255.0 {
#}
# This is a very basic subnet declaration.
#subnet 10.254.239.0 netmask 255.255.255.224 {
# range 10.254.239.10 10.254.239.20;
# option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}
# This declaration allows BOOTP clients to get dynamic addresses.
```

On va alors chercher ces lignes-ci et les modifier avec cette configuration :

```
# A slightly different configuration for an internal subnet.
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.100 192.168.10.200;
    option domain-name-servers 192.168.10.5, 192.168.10.6;
    option domain-name "carnoflux domain";
    option routers 192.168.10.254;
    option subnet-mask 255.255.255.0;
# option broadcast-address 10.5.5.31;
    default-lease-time 520;
    max-lease-time 7200;
}
```

On peut voir qu'on peut y modifier l'adresse subnet, le masque, la range des adresses utilisées pour le DHCP, le nom de domaine, l'adresse du routeur et d'autres. On peut également choisir la durée du renouvellement de bail (ici en secondes).

Après avoir enregistré la configuration, il suffit de redémarrer le service DHCP avec la commande :

```
- service isc-dhcp-server restart
```

Le service DHCP est maintenant en place. Pour essayer, on peut par exemple installer et configurer un client Windows afin qu'il se connecte au DHCP et qu'il obtienne automatiquement une adresse de la plage que l'on a configurée dans le fichier dhcp.conf (ici 192.168.10.100 – 192.168.10.200).

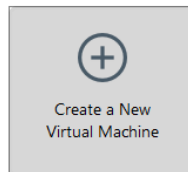
Client Windows

Nous allons finalement configurer un client sous Windows qui permettra d'avoir un accès aux sites web. Il sera intégré localement dans le réseau et une adresse IP lui sera distribuée automatiquement par notre DHCP.

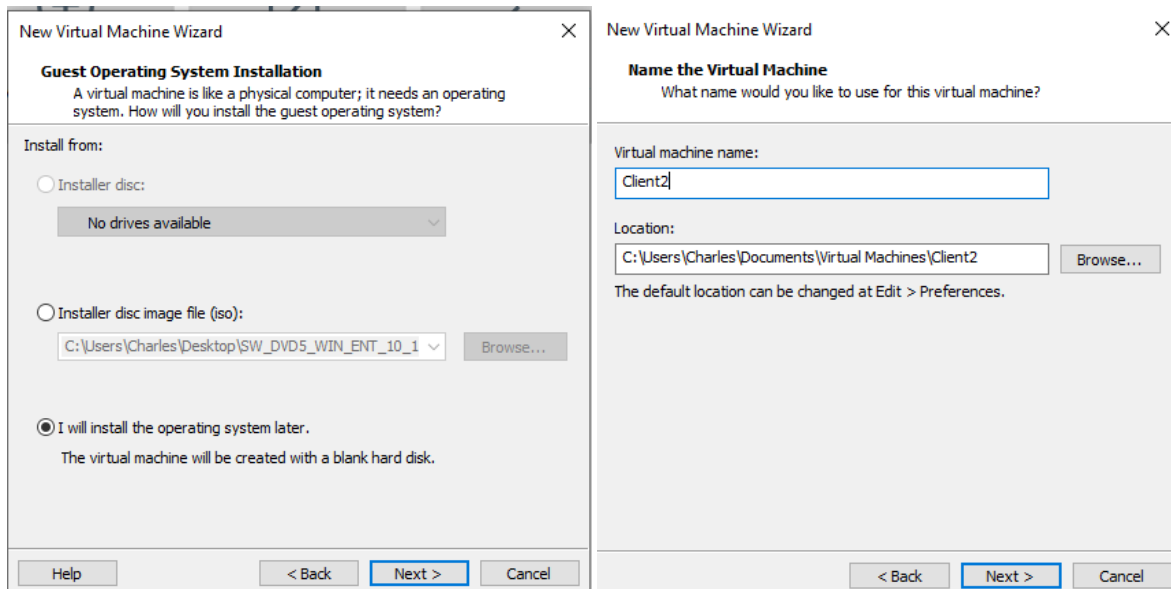
Un ISO de Windows (ici Windows10) sera à se procurer (<https://www.microsoft.com/fr-fr/software-download/windows10>)

On va ensuite installer notre OS à partir de cet ISO (utiliser une clé bootable pour installer sur un PC physique)

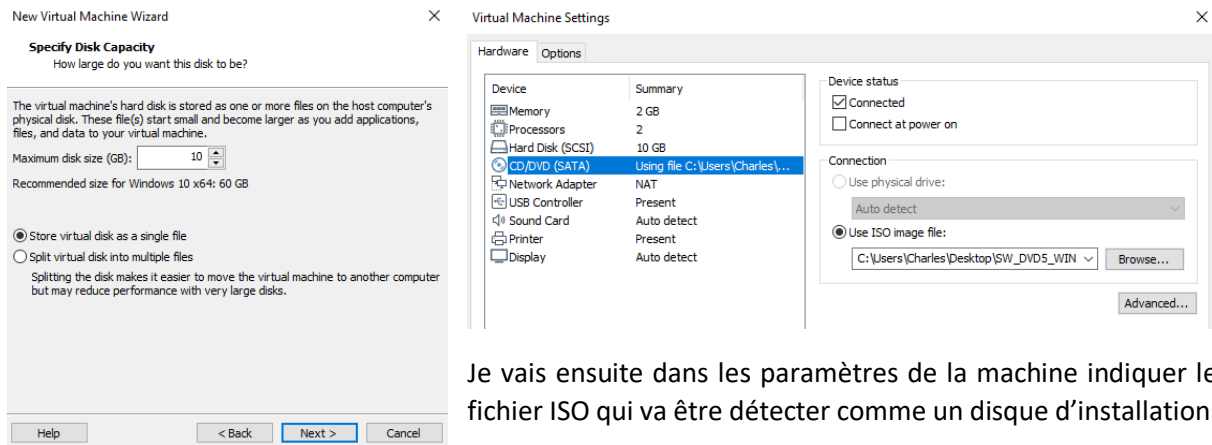
Sous VMWare : on cliquera sur ce bouton :
d'installation de VMWare et Windows :



puis nous suivrons la procédure

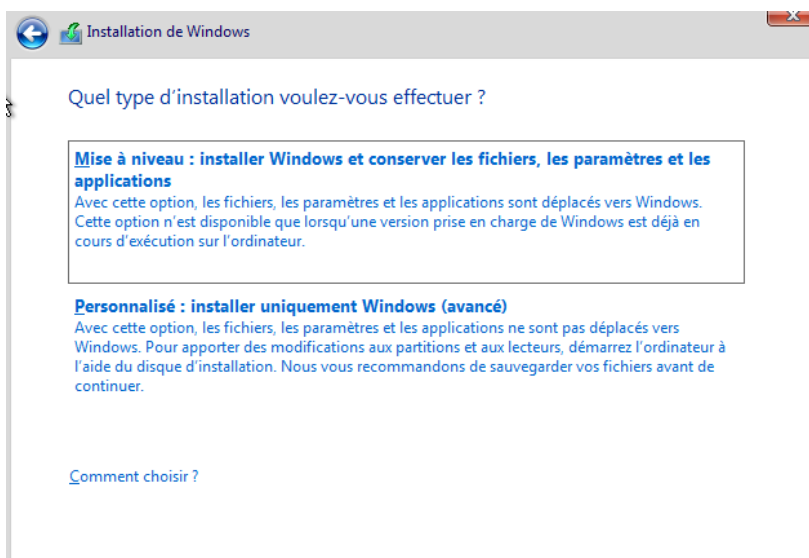
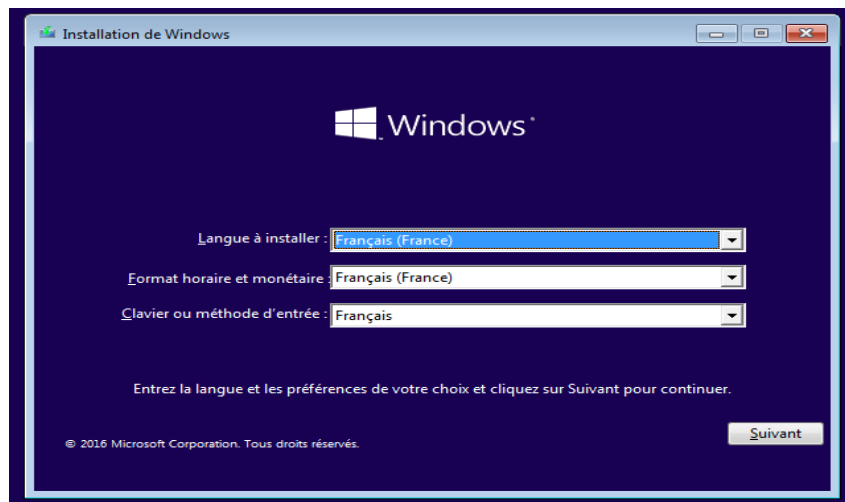


(Je donne ici « Client2 » come nom de machine virtuelle puis je choisis l'espace disque attribué)

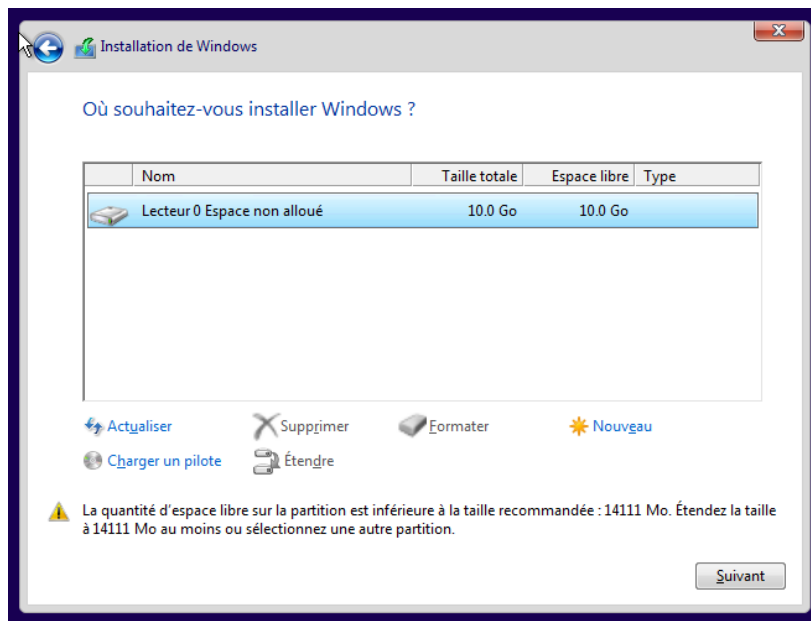


Je vais ensuite dans les paramètres de la machine indiquer le fichier ISO qui va être détecter comme un disque d'installation.

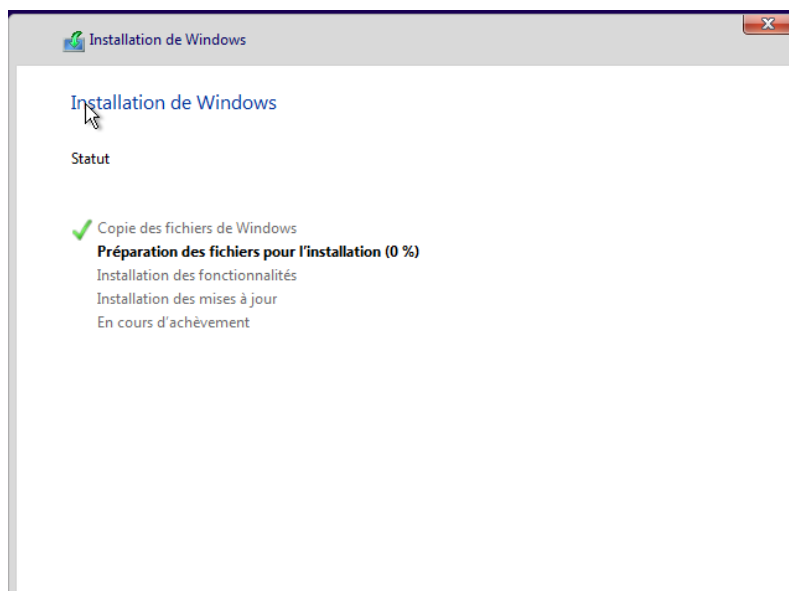
Une fois la configuration de la machine virtuelle effectuée, on va pouvoir passer à l'installation et la configuration de Windows :



Choisir « Personnalisé : installer uniquement Windows (avancé) ».

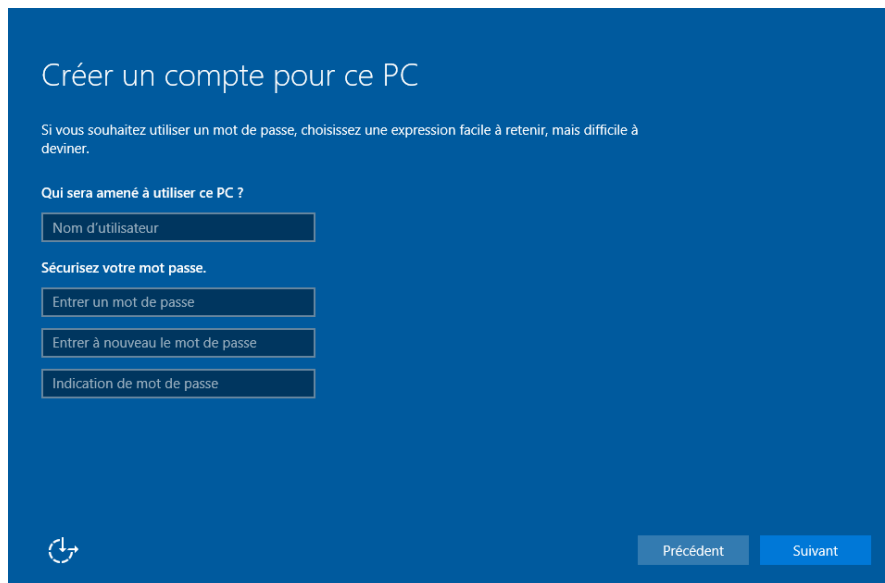


Choisir le disque pour l'installation.



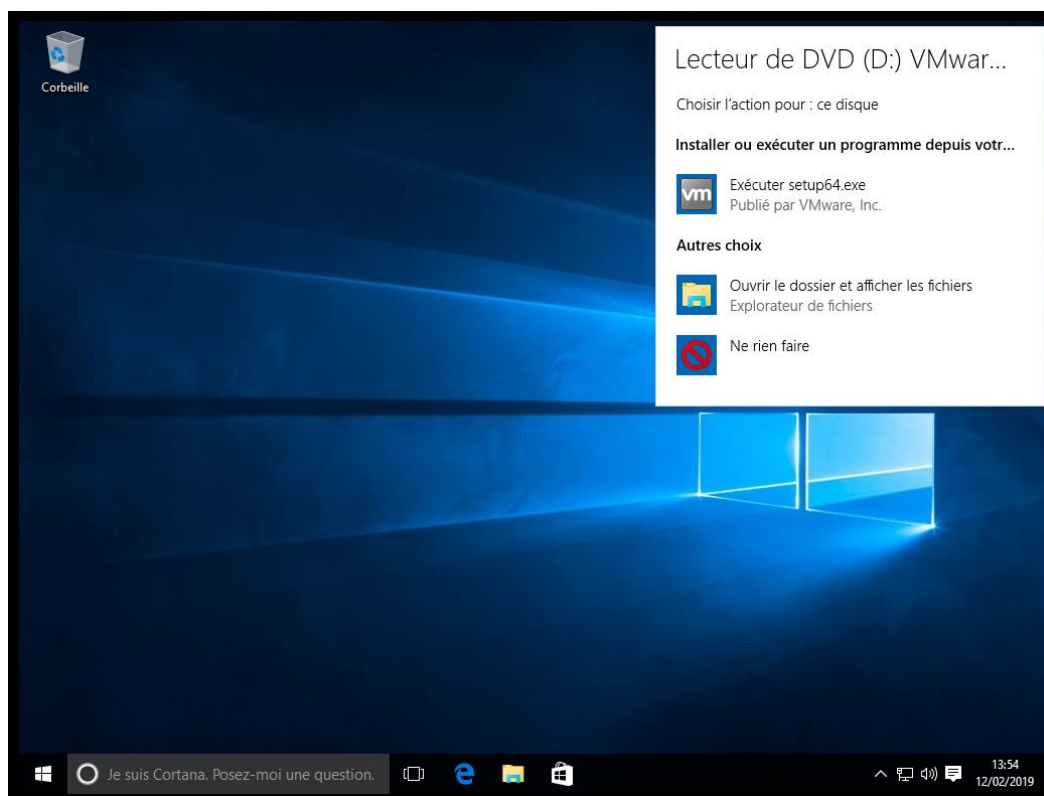
L'installation s'effectue ensuite automatiquement et le machine va redémarrer.

Windows va maintenant nous inviter à configurer notre système :



Une fois configuré, nous aurons accès au bureau.

Sur VMWare, il va être utile d'installer les « VMWare Tools » :



(Note : Windows sera utilisable pendant 30 jours si aucune clé de produit n'est rentrée)

Maintenant, la machine se situant en local sur le même réseau, on peut vérifier que l'adresse IP du client provient bien de notre DHCP :

```
C:\Windows\system32\cmd.exe
Microsoft Windows [version 10.0.10586]
(c) 2015 Microsoft Corporation. Tous droits réservés.

C:\Users\Charles>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet0 :

    Suffixe DNS propre à la connexion. . . : carnofluxe.domain
    Adresse IPv6 de liaison locale. . . . : fe80::61b3:d481:8583:4d3%6
    Adresse IPv4. . . . . : 192.168.10.103
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 192.168.10.254

Carte Ethernet Connexion réseau Bluetooth :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . :

Carte Tunnel isatap.carnofluxe.domain :

    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . : carnofluxe.domain

C:\Users\Charles>
```

On remarque que l'IP est 192.168.10.103, donc bien attribuée par le DHCP (dont la plage est de 100 à 200).

On va également maintenant pouvoir accéder à nos sites web depuis ce client si les DNS sont bien configurés :

