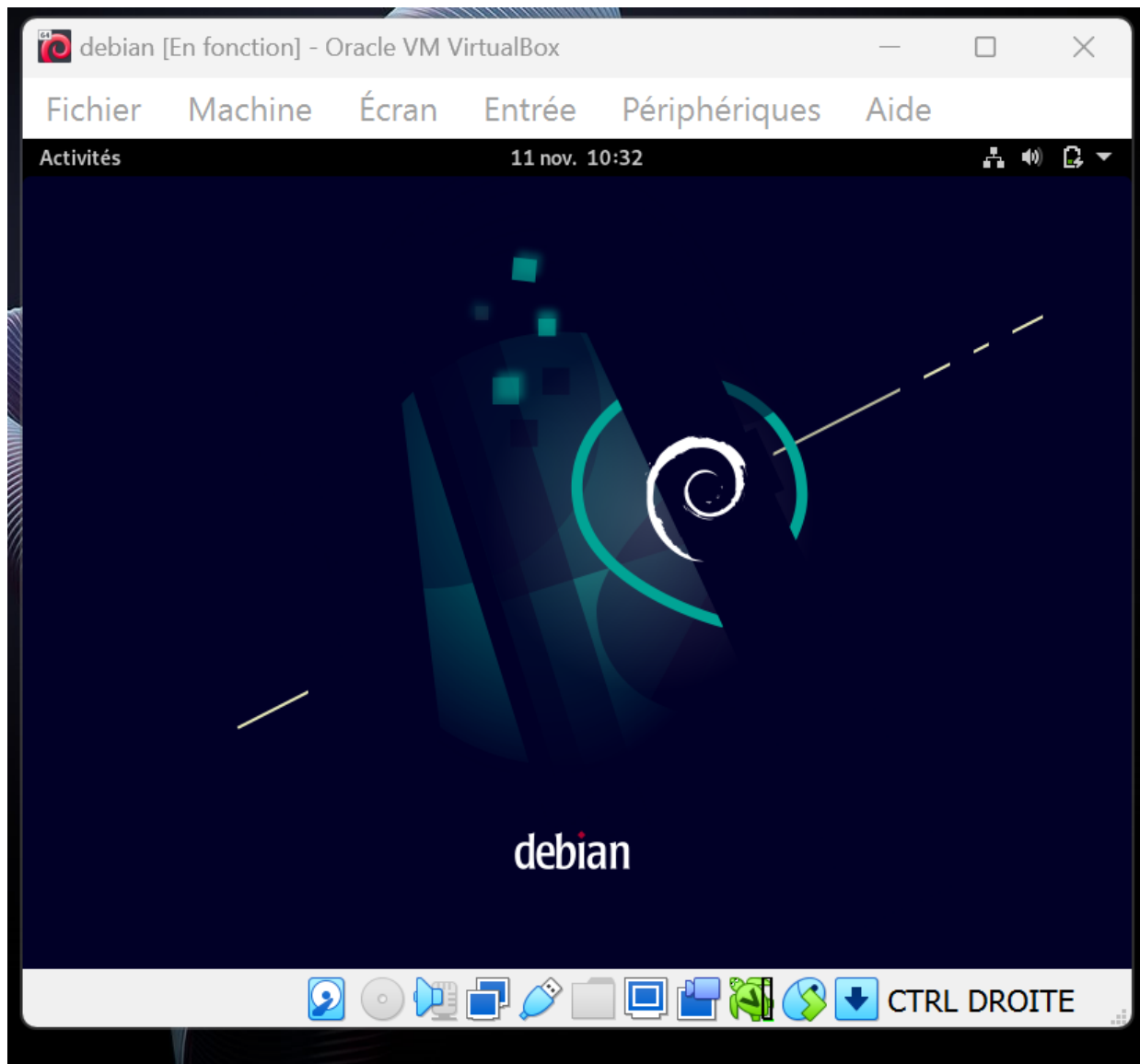
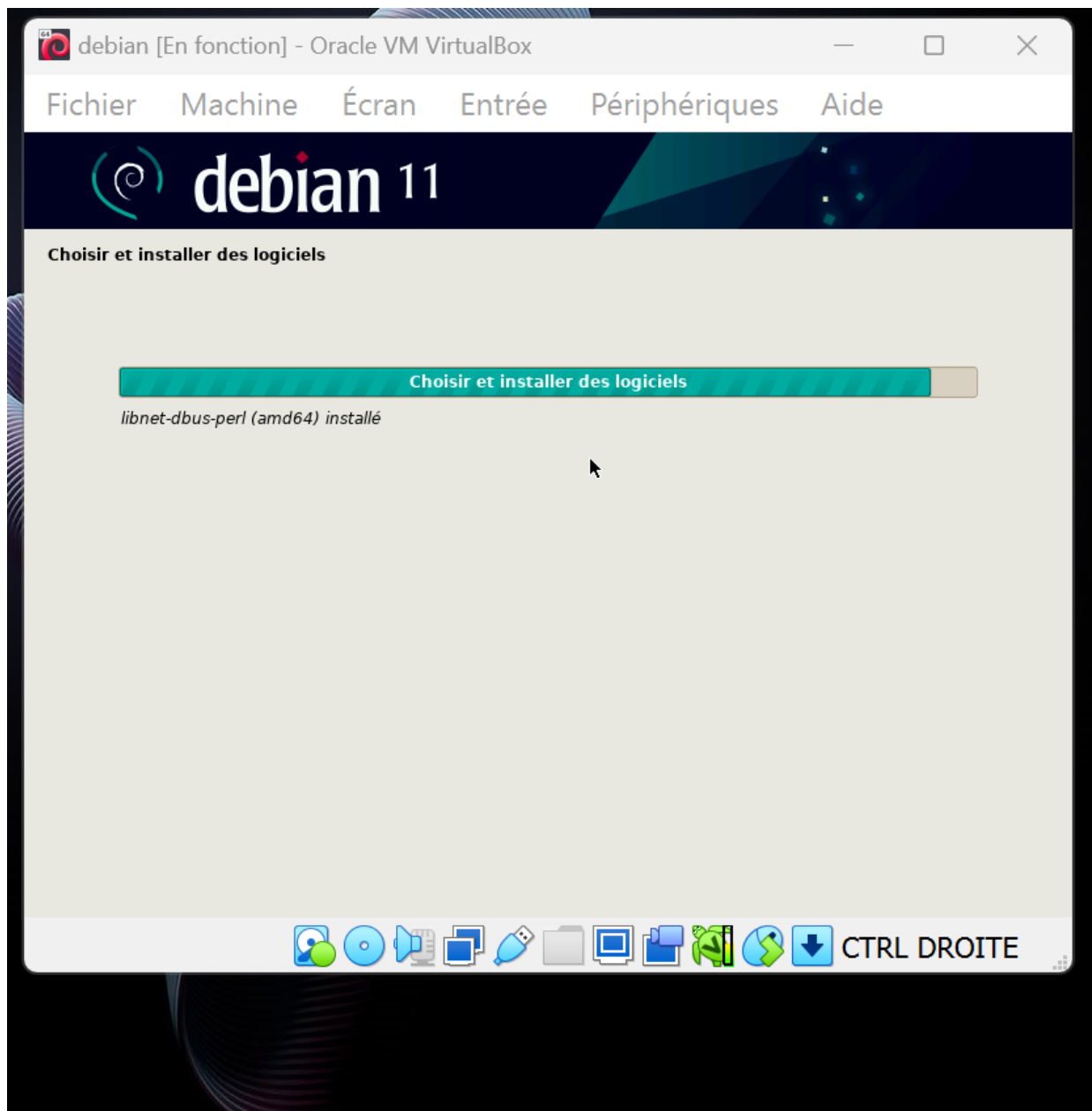


Projet DDWS

Job 1 :

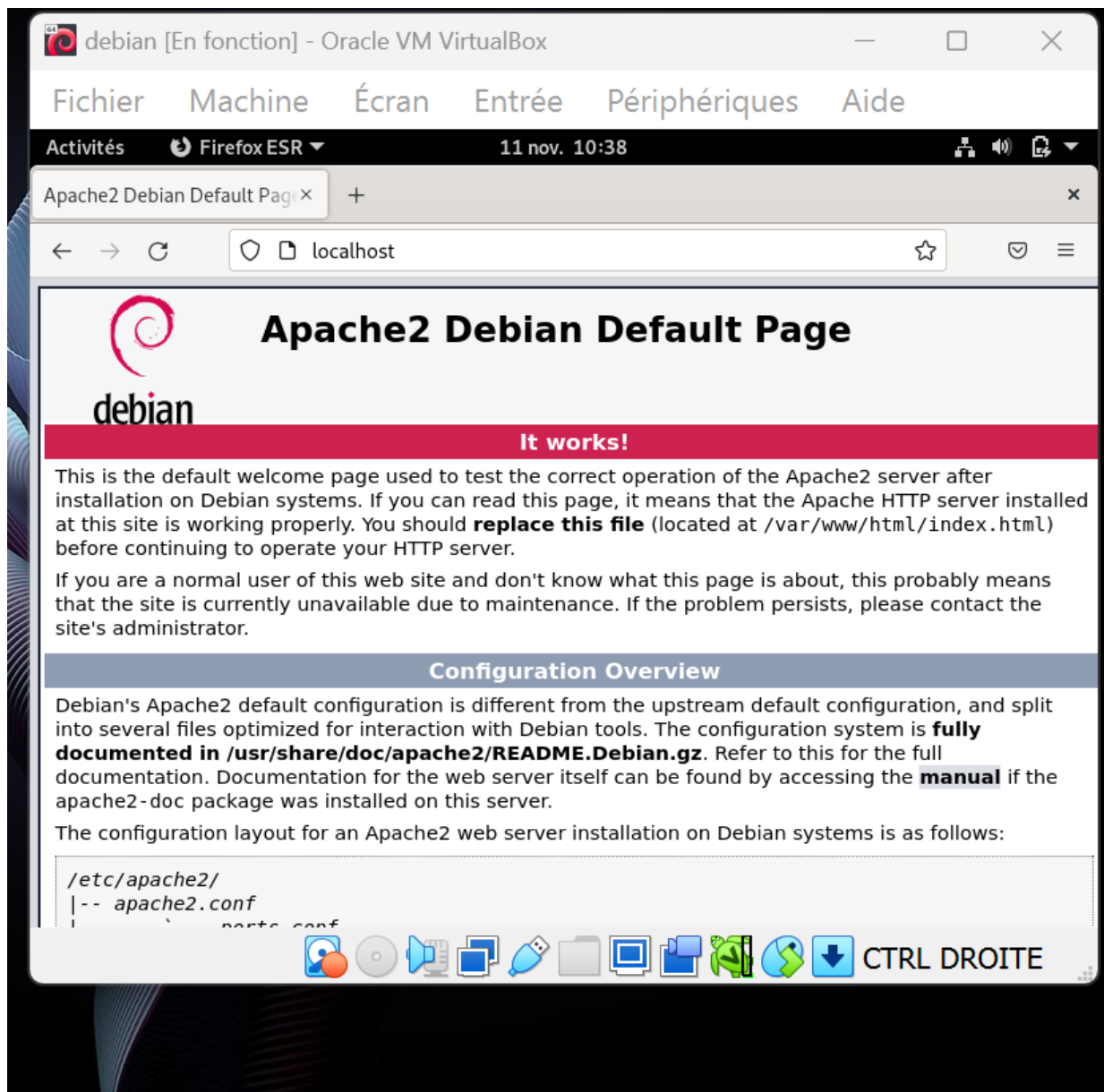
- Installer Oracle VM VirtualBox
- télécharger une image ISO Debian 11.
- installer l'OS debian 11 sur la VM





Job 2 :

- installation Shh (Sudo apt-get install openssh-server)
- Mettre à jour le système (sudo apt update && apt upgrade)
- Installer Apache2 (sudo apt install apache2)



Job 3 :

Serveur Apache, avantage,

- Open-source et gratuit même pour un usage commercial.
- Logiciel fiable et stable.
- Mise à jour régulière, correctifs de sécurité réguliers.
- Flexible grâce à sa structure basée sur des modules.
- Facile à configurer, adapté aux débutants.
- Plateforme-Cross (fonctionne sur les serveurs Unix et Windows).
- Fonctionne avec les sites WordPress.
- Grande communauté et support disponible en cas de problème.

Serveur Apache, inconvénient,

- Problèmes de performances sur les sites web avec un énorme trafic.
- Trop d'options de configuration peuvent mener à la vulnérabilité de la sécurité.

Serveur Nginx, avantage,

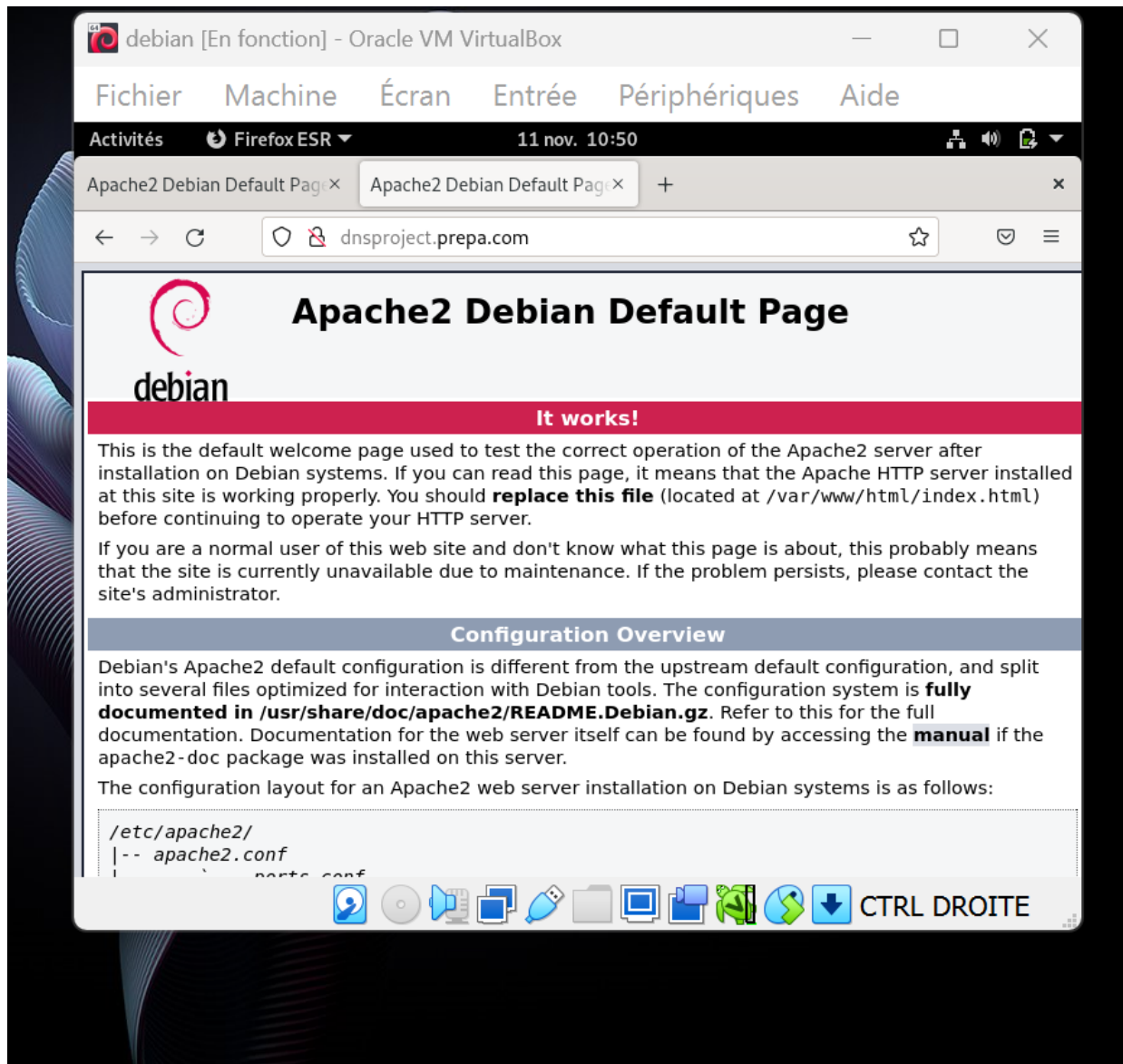
- Vitesse – Nginx sert du contenu statique environ 2,5 fois plus rapidement qu'Apache. Il s'agit là d'une grande différence de vitesse.
- S'adapte mieux qu'Apache – Nginx gère mieux le trafic élevé qu'Apache, une autre raison pour laquelle il est plus rapide.
- Nécessite moins de ressources – En raison du fonctionnement de Nginx, il nécessite moins de mémoire, ce qui peut vous aider à économiser sur les coûts d'hébergement.

Serveur Nginx, inconvénients,

- Options limitées – Peu d'hébergeurs offrent la prise en charge de Nginx, vous avez donc moins de plans à disposition sous Nginx.
- Communauté moins développée – Apache a une énorme communauté et des tonnes de modules qui facilitent l'obtention d'une assistance pour faire à peu près n'importe quoi.
- Une moins bonne option pour servir du contenu dynamique – Nginx utilise un logiciel tiers pour gérer les demandes de contenu dynamique. Dans certains cas, il peut fonctionner moins bien qu'Apache.

Job 4 :

- Dans /etc/hosts ajouter une ligne ""adresse IP" dsnproject.prepa.com" avec la commande `sudo nano /etc/hosts`
- Dans /etc/resolv.conf ajouter une ligne ""nameserveradresse IP" dsnproject.prepa.com"



Job 5 :

Le nom de domaine est l'adresse de votre site que les gens vont devoir taper dans leur navigateur web afin d'accéder à votre site Internet. Pour obtenir un nom de domaine il faut :

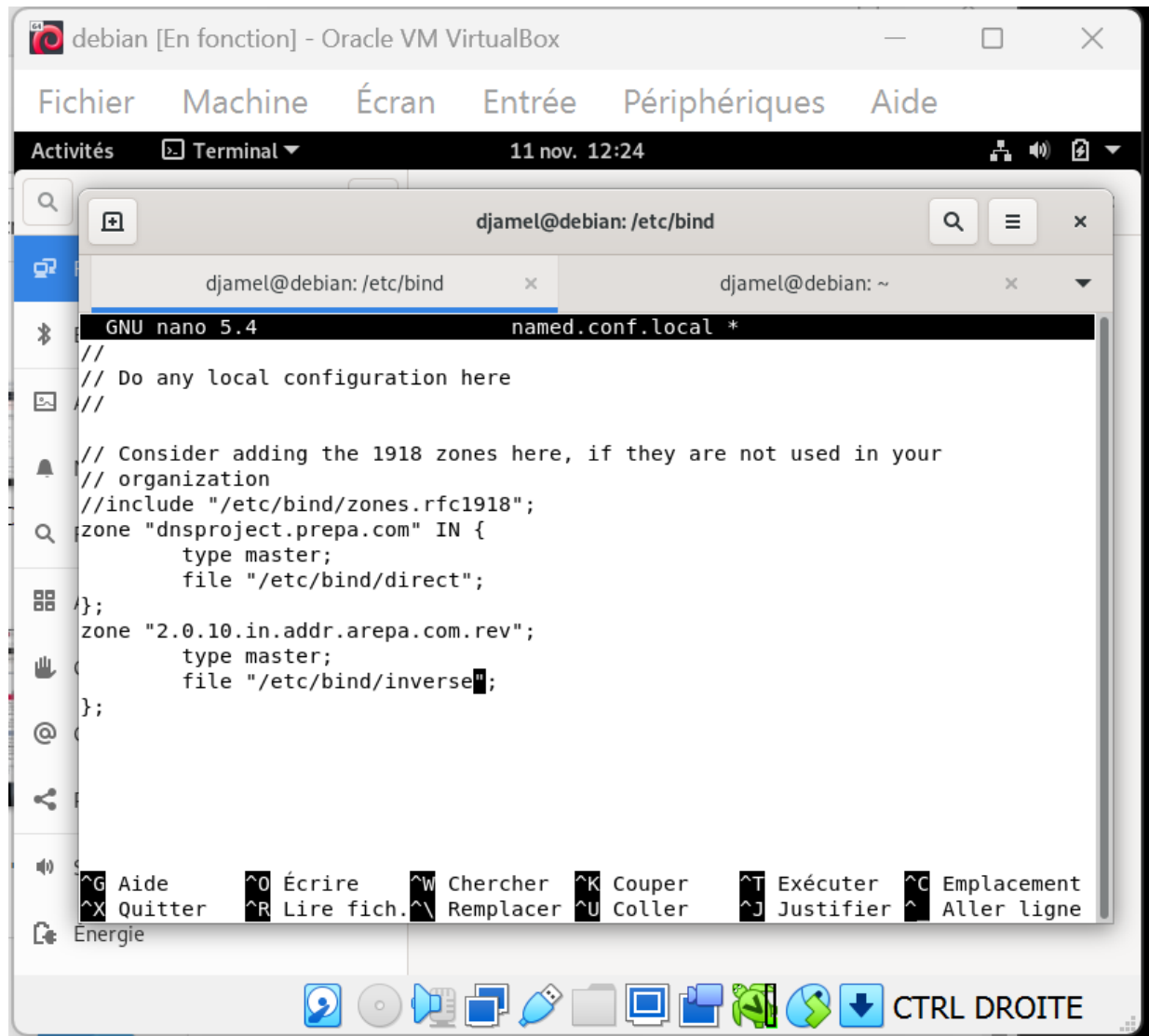
- choisir un bureau d'enregistrement de noms de domaine accrédité
- trouver le bon nom de domaine, y compris le domaine de premier niveau
- vérifier la disponibilité de l'adresse avec Domain Check
- commande ou enregistrement du nom de domaine
- vérifier la validité du nouveau domaine

Job 6 :

installer bind9 (sudo apt-get install bind9)

éditer le dossier named.conf.local

(sudo nano named.conf.local)



The screenshot shows a Debian VM window titled "debian [En fonction] - Oracle VM VirtualBox". The window has a menu bar with "Fichier", "Machine", "Écran", "Entrée", "Périphériques", and "Aide". Below the menu bar is a toolbar with "Activités", "Terminal", and a clock showing "11 nov. 12:24". The main area displays a terminal window with the prompt "djamel@debian: /etc/bind". The terminal shows the GNU nano 5.4 editor editing the file "named.conf.local". The content of the file is as follows:

```
//  
// 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 "dnsproject.prepa.com" IN {  
    type master;  
    file "/etc/bind/direct";  
};  
zone "2.0.10.in.addr.arepa.com.rev";  
    type master;  
    file "/etc/bind/inverse";  
};
```

At the bottom of the terminal window, there is a keyboard shortcut table:

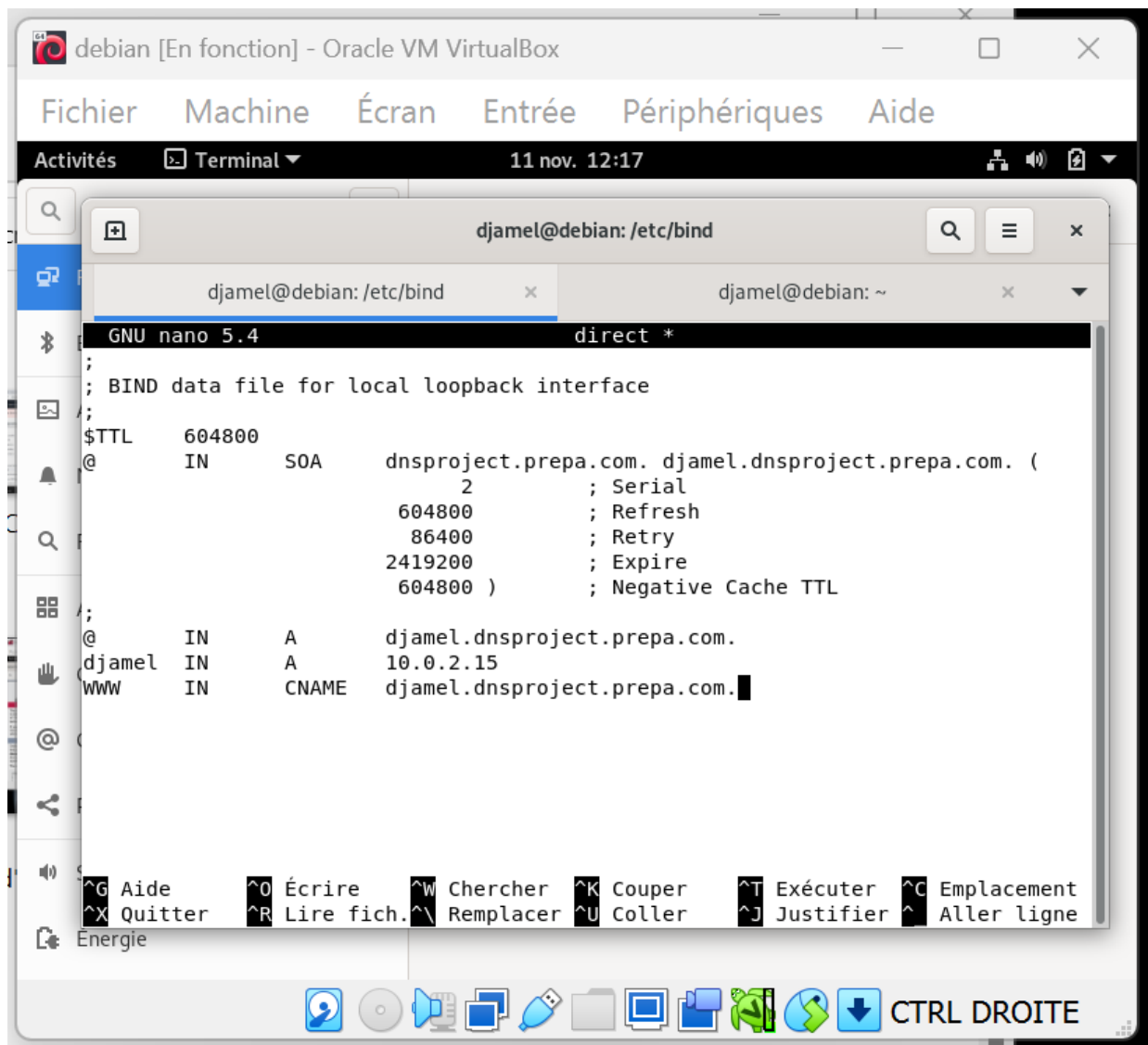
^G Aide	^O Écrire	^W Chercher	^K Couper	^T Exécuter	^C Emplacement
^X Quitter	^R Lire fich.	^_ Remplacer	^U Coller	^J Justifier	^_ Aller ligne

The bottom of the window shows a taskbar with various icons and the text "CTRL DROITE".

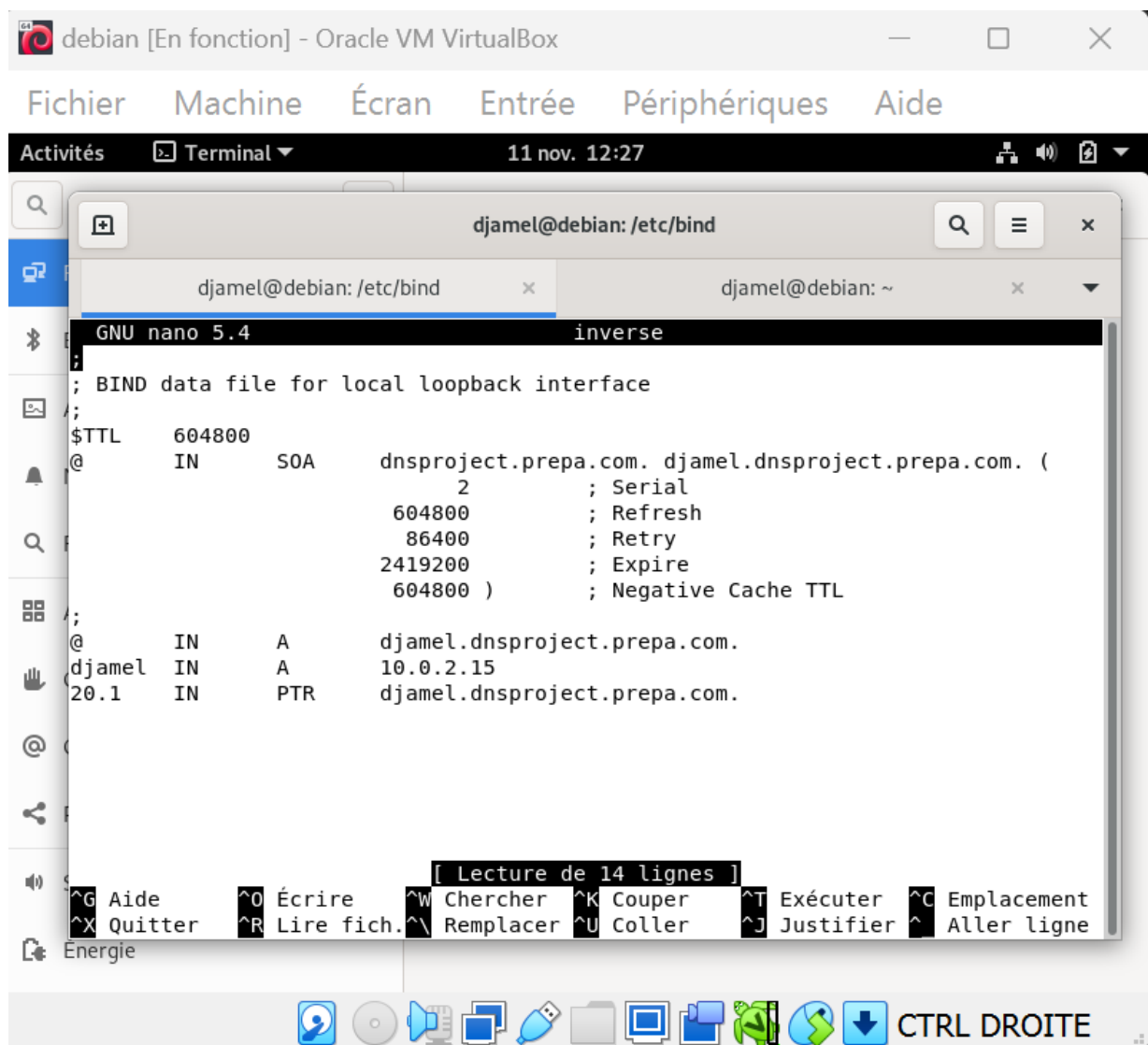
éditer le fichier direct

(sudo cp db.local direct)

(sudo nano direct)



editer le fichier inverse
(sudo cp direct inverse)
(sudo nano inverse)



ensuite restart le service bind9
(sudo service bind9 restart)

Job 7 :

- apt-get install isc-dhcp-server
- Sauvegarde du fichier de configuration par défaut : cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf-bak
- Édition du fichier de configuration du service : nano /etc/dhcp/dhcpd.conf ajouter à la fin :

```

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.2 192.168.1.50;
    option domain-name-servers 192.168.1.1;
    option domain-name
"dnsproject.prepa.com"; option
netbios-name-servers 192.168.1.1;
    option routers 192.168.1.1;
    option subnet-mask 255.255.255.0;
}

```



```
option broadcast-address 192.168.1.255;  
default-lease-time 86400;  
max-lease-time 676800;  
}
```

- redémarrer : `systemctl restart isc-dhcp-serve`

Job 8 : (en NAT)

- sur nano /etc/network/interfaces :

```
auto lo
```

```
# The loopback network
interface iface lo inet loopback
#carte vers la box
auto ens33
iface ens33 inet
static address
192.168.0.1
network 192.168.0.0
netmask 255.255.255.0
gateway 192.168.0.254
#post-up iptables-restore < /etc/iptables.save # ( ligne à décommenter plus tard )
```

```
#carte vers le LAN
auto ens34
iface ens34 inet
static address
192.168.1.1
network 192.168.1.0
netmask 255.255.255.0
```

- prise en compte des modification : /etc/init.d/networking start
- modification du fichier /etc/sysctl.conf pour dé-commenter :

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

- prise en compte des modification : sysctl -p

- Mise en place du protocole NAT sur ens33

```
iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE
```

- Sauvegarde de cette modification :

```
iptables-save > /etc/iptables.save
```

- Édition de /etc/network/interfaces

Décommenter la ligne suivante :

```
post-up iptables-restore < /etc/iptables.save
```

- Au prochain démarrage du système :

```
iptables -L -t nat
```

- sur les VM reliés à la passerelle ajouter en Gateway 192.168.1.1

Job 9 :

- installer le pare-feu : `apt-get update && apt install ufw`
- On met en route le pare-feu : `ufw enable`
- Pour interdire le ping il faut commenter la ligne suivante dans le fichier `/etc/ufw/before.rules` :
 `-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT`
- appliquer les changements : `ufw disable && ufw enable`

Job 10 :

- installer Samba : `apt install samba`
- éditer fichier de configuration Samba `/etc/samba/smb.conf`
: [partage]
 comment = Partage de données
 path = /srv/partage
 guest ok = no
 read only = no
 browseable =
 yes
 valid users = @partage
- on redémarre samba : `systemctl restart smbd*`
- Le groupe "partage" que nous avons déclaré dans la configuration n'existe pas. Nous allons créer le groupe avec la commande : `groupadd partage`
- créer les divers utilisateurs avec la commande : `adduser`
- lui attribuer un mot de passe avec la commande : `smbpasswd -a "username"`
- ajouter les utilisateurs aux groupes avec la commande : `gpasswd -a "username" partage`
- Le partage sera hébergé à l'emplacement `"/srv/partage"` du serveur.
- Créer le dossier : `mkdir /srv/partage`
- groupe "partage" comme propriétaire du dossier : `chgrp -R partage /srv/partage/`
- ajouter droit écriture et lecture au groupe : `chmod -R g+rw /srv/partage/`

Pour aller plus loin :

- Activer le module SSL : `a2enmod ssl`
- Activer le module headers : `a2enmod header` (le module headers permet d'activer la directive HSTS.)
- redémarrer le service : `systemctl restart apache2`
- Création du répertoire qui contiendra le certificat SSL : `mkdir /etc/ssl/dnsproject.prepa.com/`
- Déclarer VirtualHosts pour le HTTPS; pour accéder à votre site en HTTPS il faut créer un VirtualHost dédié, pour cela :
 Dans `/etc/apache2/sites-available` : dupliquer fichier de configuration "000-default.conf" en "000-default-ssl.conf" avec :
 `cp 000-default.conf 000-default-ssl.conf`
 Modifiez le fichier dupliqué, remplacez le port 80 du VirtualHost par le port 443 : `<VirtualHost *:80> → <VirtualHost *:443>`

Ajoute les instructions suivantes avant le

```
</VirtualHost> SSLEngine On
```

```
SSLProtocol All -SSLv3 -SSLv2
```

```
SSLCipherSuite HIGH:!aNULL:!MD5:!ADH:!RC4:!DH
```

```
SSLHonorCipherOrder on
```

```
SSLCertificateFile "/etc/ssl/votre-domaine-fr/www.dnsproject.prepa.com.cer"
```

```
SSLCertificateKeyFile
```

```
"/etc/ssl/votre-domaine-fr/www.dnsproject.prepa.com.key"
```

```
Header always set Strict-Transport-Security "max-age=15768000"
```

- activer VirtualHost : a2ensite nom-du-vhost-ssl
- redémarrer le service : systemctl restart apache2

Lorsque vous achetez un certificat SSL “traditionnel”, vous savez qu'il a été signé par une autorité de certification réputée. En revanche, un certificat auto-signé n'est pas signé par une autorité comme SSL ou TLS ; il est créé, mis en œuvre et signé par un développeur de logiciels tiers.

Le certificat est auto-signé, il n'est donc pas reconnu comme étant suffisamment sécurisé.