



DDWS

Date : 25/10/2023

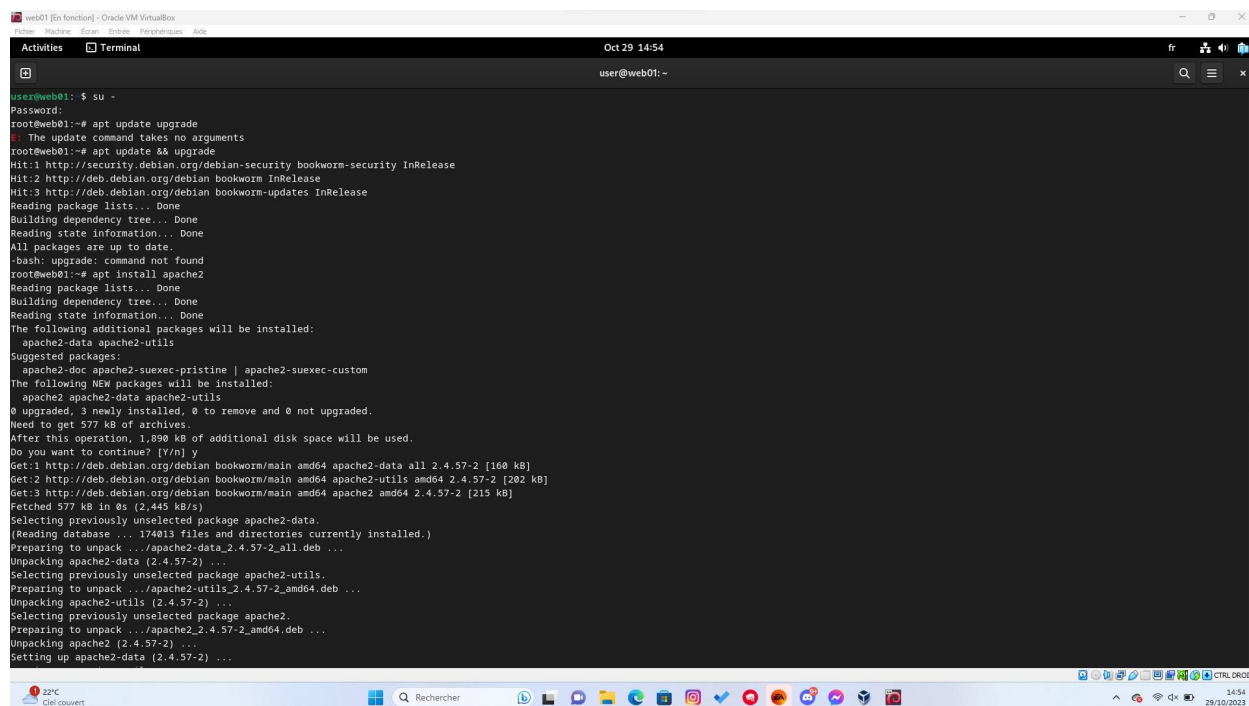
Job 01 : Pour ce sujet, il vous faudra installer une VM Debian, avec interface graphique.

- Installer OpenSSH en tant que root avec la commande "apt-get install openssh-server".
- Editer le fichier sshd_config avec la commande "nano /etc/ssh/sshd_config".
- Changer le port d'écoute en décommentant la ligne "#Port 22".
- Désactiver la connexion root en éditant le fichier sshd_config.
- Configurer l'authentification SSH basée sur les clés.
- Redémarrer le serveur SSH pour appliquer les modifications.

[How To Install and Enable SSH Server on Debian 10 \(devconnected.com\)](https://devconnected.com/how-to-install-and-enable-ssh-server-on-debian-10/)

Job 02 : Pour commencer, installer un serveur Web Apache2.

[How to Install Apache2 Web Server on Debian 12 – LinuxWays](#)



```

user@web01:~$ su -
root@web01:~# apt update upgrade
The update command takes no arguments
root@web01:~# apt update && upgrade
Hit:1 http://security.debian.org/debian-security bookworm-security InRelease
Hit:2 http://deb.debian.org/debian bookworm InRelease
Hit:3 http://deb.debian.org/debian bookworm-updates InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
-bash: upgrade: command not found
root@web01:~# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-data apache2-utils
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-data apache2-utils
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 577 kB of archives.
After this operation, 1,890 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bookworm/main amd64 apache2-data all 2.4.57-2 [160 kB]
Get:2 http://deb.debian.org/debian bookworm/main amd64 apache2-utils amd64 2.4.57-2 [282 kB]
Get:3 http://deb.debian.org/debian bookworm/main amd64 apache2 amd64 2.4.57-2 [215 kB]
Fetched 577 kB in 6s (2,445 kB/s)
Selecting previously unselected package apache2-data.
(Reading database ... 174013 files and directories currently installed.)
Preparing to unpack .../apache2-data_2.4.57-2_all.deb ...
Unpacking apache2-data (2.4.57-2) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../apache2-utils_2.4.57-2_amd64.deb ...
Unpacking apache2-utils (2.4.57-2) ...
Selecting previously unselected package apache2.
Preparing to unpack .../apache2_2.4.57-2_amd64.deb ...
Unpacking apache2 (2.4.57-2) ...
Setting up apache2-data (2.4.57-2) ...

```

A l'aide de la commande : `apt install apache2`

Job 03 : Renseignez-vous sur les différents serveurs Web existants

Il existe de nombreux serveurs web disponibles, chacun ayant ses propres caractéristiques, avantages et inconvénients. Voici une liste de certains des serveurs web les plus populaires :

Apache HTTP Server (httpd) : Apache est l'un des serveurs web les plus anciens et les plus populaires. Il est open source et largement utilisé dans le monde entier. Il prend en charge de nombreux modules et fonctionnalités personnalisables.

Nginx : Nginx est un serveur web et proxy inverse réputé pour sa haute performance et sa faible utilisation des ressources. Il est souvent utilisé pour servir du contenu statique, équilibrer la charge et servir de proxy.

Microsoft Internet Information Services (IIS) : IIS est le serveur web de Microsoft pour les systèmes Windows. Il offre une intégration étroite avec les technologies Microsoft telles que ASP.NET.

des besoins spécifiques de votre projet, de vos compétences **techCaddy** : Caddy est un serveur web open source qui se distingue par sa facilité de configuration et son

LiteSpeed Web Server : LiteSpeed est connu pour sa vitesse élevée et sa faible utilisation des ressources. Il est souvent utilisé dans les environnements d'hébergement web.

Ces serveurs web varient en termes de performances, de fonctionnalités, de langages de programmation pris en charge et de complexité de configuration. Le choix du serveur web dépendra support intégré du chiffrement SSL.

Cherokee : Cherokee est un serveur web open source qui propose une interface web conviviale pour la configuration. Il est axé sur la facilité d'utilisation.

Lighttpd (Lighty) : Lighttpd est un serveur web léger et rapide conçu pour les environnements où la vitesse et l'efficacité sont essentielles.

Tomcat (Apache Tomcat) : Tomcat est un conteneur de servlet Java largement utilisé pour exécuter des applications Java EE. Il est souvent utilisé pour des applications Java, y compris des applications web.

CherryPy : CherryPy est un serveur web Python qui permet de développer des applications web Python de manière simple et élégante.

Node.js (avec Express.js) : Node.js est une plateforme qui permet d'exécuter du code JavaScript côté serveur. Express.js est un framework couramment utilisé pour la création de serveurs web avec Node.js.

Gunicorn : Gunicorn est un serveur web WSGI (Web Server Gateway Interface) pour les applications Python. Il est souvent utilisé avec des applications basées sur Python.

Chacun des serveurs web mentionnés précédemment a ses avantages et ses inconvénients. Le choix du serveur web dépendra de vos besoins spécifiques et de la plateforme sur laquelle vous travaillez. Voici un aperçu des avantages et des inconvénients de certains des serveurs web populaires :

Apache HTTP Server :

- Avantages :
 - Vaste base d'utilisateurs et de documentation.
 - Hautement configurable grâce à des modules.
- Inconvénients :
 - Peut être complexe à configurer pour les débutants.
 - La configuration par défaut peut être gourmande en ressources.

Nginx :

- Avantages :
 - Hautes performances et faible utilisation des ressources.
 - Prise en charge de la mise en cache et de l'équilibrage de charge.
- Inconvénients :
 - La configuration peut sembler complexe pour les débutants.
 - Moins de modules tiers par rapport à Apache.

Microsoft Internet Information Services (IIS) :

- Avantages :
 - Intégration étroite avec les technologies Microsoft.
 - Facile à configurer pour les environnements Windows.

- Inconvénients :
 - Limité aux systèmes Windows.
 - Moins populaire dans les environnements non-Windows.

LiteSpeed Web Server :

- Avantages :
 - Haute performance.
 - Utilisation efficace des ressources.
- Inconvénients :
 - Souvent soumis à des licences commerciales.

Caddy :

- Avantages :
 - Facilité de configuration avec support SSL intégré.
 - Convivial pour les débutants.
- Inconvénients :
 - Certaines fonctionnalités avancées peuvent nécessiter une licence commerciale.

Cherokee :

- Avantages :
 - Interface web conviviale pour la configuration.
 - Prise en charge de plusieurs protocoles.
- Inconvénients :
 - Moins populaire que certains autres serveurs web.

Lighttpd :

- Avantages :
 - Léger et rapide.
 - Faible utilisation des ressources.
- Inconvénients :
 - Moins de fonctionnalités avancées que d'autres serveurs.

Tomcat (Apache Tomcat) :

- Avantages :
 - Conçu pour exécuter des applications Java.
 - Prise en charge de servlets Java EE.
- Inconvénients :
 - Convient principalement aux applications Java.

CherryPy :

- Avantages :
 - Facilité de développement d'applications web Python.
 - Framework Pythonic.
- Inconvénients :
 - Moins de fonctionnalités par rapport aux serveurs web complets.

Node.js (avec Express.js) :

- Avantages :
 - Haute performance pour les applications basées sur JavaScript.
 - Évolutif et non bloquant.
- Inconvénients :
 - Convient principalement aux applications JavaScript.

Gunicorn :

- Avantages :
 - Prise en charge de l'exécution d'applications Python via WSGI.
 - Évolutif.
- Inconvénients :
 - Ne gère que des applications Python.

Le choix d'un serveur web dépendra de facteurs tels que la performance, la facilité de configuration, la prise en charge des langages de programmation, la taille de la communauté et les exigences spécifiques de votre projet. Il est recommandé de faire des recherches approfondies et de tester plusieurs serveurs web pour déterminer celui qui convient le mieux à votre cas d'utilisation.

Job 04 : Mettez en place un DNS sur votre serveur Linux qui fera correspondre l'adresse IP de votre serveur au nom de domaine local suivant : "dnsproject.prepa.com"

[debian - How to set the domain name on GNU/Linux? - Server Fault](#)

[Configurer Bind9 sur Ubuntu, Debian - malekal.com](#)

Dans un premier temps nous installons bind9 ainsi que ses utilitaires à l'aide de la commande : `sudo apt install -y bind9 bind9utils bind9-doc dnsutils`

Ensuite nous passons la machine virtuelle en bridge dans les paramètres de connection de la machine virtuelle.

j'utilise ensuite la commande `ip a` pour connaître mon ip

```

user@web01:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b8:70:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.32/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 84813sec preferred_lft 84813sec
    inet6 2a01:cb1c:57f:e000:4c6e:a200:f57e:90d1/64 scope global temporary deprecated dynamic
        valid_lft 85505sec preferred_lft 0sec
    inet6 2a01:cb1c:57f:e000:a00:27ff:feb8:701c/64 scope global deprecated dynamic mngtmpaddr noprefixroute
        valid_lft 85505sec preferred_lft 0sec
    inet6 fe80::a00:27ff:feb8:701c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Mon adresse ip est : **192.168.1.32**

je me déplace dans les fichiers de bind qui se trouve dans le répertoire **/etc/bind**

je modifie ensuite les fichiers db.local, named.conf.local, resolv.conf de la manière suivante :

```

$TTL      604799
@          IN      SOA      prepa.com dnsproject.prepa.com. (
                           2          ; Serial
                           604800     ; Refresh
                           86400      ; Retry
                           2419200    ; Expire
                           604800 )   ; Negative Cache TTL
;
@          IN      NS       dnsproject.prepa.com.
dnsproject IN      A        192.168.1.32
www        IN      CNAME    dnsproject.prepa.com.

```

modification de db.local et ensuite création du fichier inverse


```
$TTL      604800
@         IN      SOA      prepa.com dnsproject.prepa.com. (
                        2      ; Serial
                        604800 ; Refresh
                        86400  ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       dnsproject.prepa.com.
dnsproject IN      A        192.168.1.32
32        IN      PTR      dnsproject.prepa.com.
```

```
//
// Do any local configuration here
//
zone "prepa.com" IN {
    type master;
    file "/etc/bind/direct";
};
zone "1.168.192.in-addr-arpa" IN {
    type master;
    file "etc/bind/inverse";
};
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
```

modification du fichier named.conf.local.

```
# Generated by NetworkManager
search prepa.com
nameserver 192.168.1.32
nameserver 2a01:cb1c:57f:e000:aecf:7bff:fe11:e7a0
```

ensuite nous faisons la création d'un script pour empêcher networkmanager de modifier le fichier resolv.conf

```
vim /etc/NetworkManager/dispatcher.d/99-dns
```

- Dans lequel nous éditons le script suivant :

```
#!/bin/sh
```

```
echo "domain mondomaine.hyp" > /etc/resolv.conf
```

```
echo "search mondomaine.hyp" >> /etc/resolv.conf
```

```
echo "nameserver 192.168.0.14" >> /etc/resolv.conf
```

```
echo "#nameserver 212.27.40.240" >> /etc/resolv.conf
```

```
echo "#nameserver 212.27.40.241" >> /etc/resolv.conf
```

```
chmod 755 /etc/NetworkManager/dispatcher.d/99-dns
```

- Exécution du script :

```
bash /etc/NetworkManager/dispatcher.d/99-dns
```

Job 05 : Faites des recherches sur comment obtient-on un nom de domaine public ?

Pour obtenir un nom de domaine public, vous devez passer par un registraire de domaine accrédité. Voici les étapes générales pour obtenir un nom de domaine public :

Choisir un Nom de Domaine :

- Sélectionnez un nom de domaine qui reflète votre site web, votre entreprise ou votre projet. Il doit être unique et ne pas être déjà enregistré par quelqu'un d'autre.

Vérifier la Disponibilité :

- Utilisez les services de recherche de disponibilité de noms de domaine fournis par les registraires pour vérifier si le nom de domaine que vous avez choisi est disponible.

Choisir une Extension de Domaine :

- Choisissez l'extension de domaine (gTLD - Generic Top-Level Domain ou ccTLD - Country Code Top-Level Domain) que vous souhaitez utiliser, comme .com, .net, .org, .fr, .uk, etc.

Sélectionner un Registraire :

- Choisissez un registraire de domaine accrédité. Il existe de nombreux registraires, y compris des sociétés populaires telles que GoDaddy, Namecheap, Google Domains, etc.

Enregistrement :

- Créez un compte avec le registraire de domaine de votre choix.
- Saisissez les informations requises, y compris le nom de domaine que vous souhaitez enregistrer.
- Procédez au paiement pour l'enregistrement du nom de domaine. La durée de l'enregistrement varie généralement de 1 an à plusieurs années, en fonction de votre choix.

Informations WHOIS :

- Fournissez des informations d'inscription précises. Celles-ci sont généralement enregistrées dans une base de données publique appelée le système WHOIS.

Confirmation :

- Vous recevrez une confirmation de l'enregistrement de votre nom de domaine une fois la transaction terminée.

Configuration DNS :

- Configurez les enregistrements DNS pour votre nom de domaine, tels que les enregistrements A, MX, CNAME, etc., pour diriger le trafic vers l'adresse IP de votre site web ou de vos serveurs de messagerie.

Concernant les spécificités des extensions de nom de domaine (TLDs - Top-Level Domains) :

gTLDs (Generic Top-Level Domains) : Ce sont des extensions de domaine génériques qui ne sont pas spécifiques à un pays ou une région. Exemples de gTLDs : .com, .org, .net, .info, .io. Ils sont largement utilisés dans le monde entier et ne sont pas liés à une géographie particulière.

ccTLDs (Country Code Top-Level Domains) : Ce sont des extensions de domaine spécifiques à un pays ou une région. Exemples de ccTLDs : .us (États-Unis), .fr (France), .uk (Royaume-Uni), .de (Allemagne). Ils sont souvent utilisés pour des sites web ciblant une audience spécifique dans un pays ou une région donnée.

gTLDs Spécialisés : Il existe également des gTLDs spécialisés qui sont conçus pour des utilisations spécifiques. Par exemple, .edu est réservé aux institutions éducatives, .gov est réservé au gouvernement des États-Unis, .museum est réservé aux musées, etc.

Nouveaux gTLDs : L'ICANN (Internet Corporation for Assigned Names and Numbers) a introduit de nombreux nouveaux gTLDs ces dernières années. Cela a élargi les options de domaine au-delà des gTLDs classiques, offrant des choix tels que .app, .blog, .guru, .tech, etc.

Lorsque vous choisissez une extension de domaine, il est important de tenir compte de la pertinence pour votre site web, de la disponibilité et parfois des restrictions spécifiques imposées par certains registres de domaine pour certaines extensions.

Job 06 : Connectez votre hôte au nom de domaine local de votre serveur, pour que votre page apache soit accessible via ce même nom de domaine.



Job 07 : Mettez en place un pare-feu en utilisant ufw

On installe ufw avec la commande : `sudo apt install ufw`

Puis on modifie le fichier before.rules :

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j DROP
-A ufw-before-input -p icmp --icmp-type time-exceeded -j DROP
-A ufw-before-input -p icmp --icmp-type parameter-problem -j DROP
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP
```

On modifie la section suivante du fichier en remplaçant les **ACCEPT** par **DROP** . Puis nous ajoutons les règles suivantes avec les commandes :

```

root@web01:/etc/ufw# fw allow 80/tcp
-bash: fw: command not found
root@web01:/etc/ufw# ufw allow 80/tcp
Rule added
Rule added (v6)
root@web01:/etc/ufw# ufw allow 443/tcp
Rule added
Rule added (v6)
root@web01:/etc/ufw# ufw status
Status: active

To Action From
--
80/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
80/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)

```

Job 08 : Mettez en place sur votre serveur un dossier partagé

Commande pour créer un dossier ,dans un dossier “mnt” qui nous permettra de faire communiquer nos dossier par la suite.

```
root@web01 : /mnt/partage# sudo mkdir /mnt/partage
```

Commande pour faire le lien entre le dossier “Hôte” et le dossier “VM”.

```
root@web01 : /home/user# sudo mount -t vboxsf Rep VM
/mnt/partage
```

“Rep_VM” représente le chemin. Et /mnt/”partage” le dossier créé précédemment.

Depuis votre gestionnaire de fichier Hôte copier un fichier dans le dossier partager créé plus tôt sur votre hôte. Une fois copier il sera aussi présent sur la VM, et inversement

sur l'on crée un fichier depuis la VM dans le fichier "partagé" créé précédemment , il sera présent dans votre dossier de partage hôte.

