

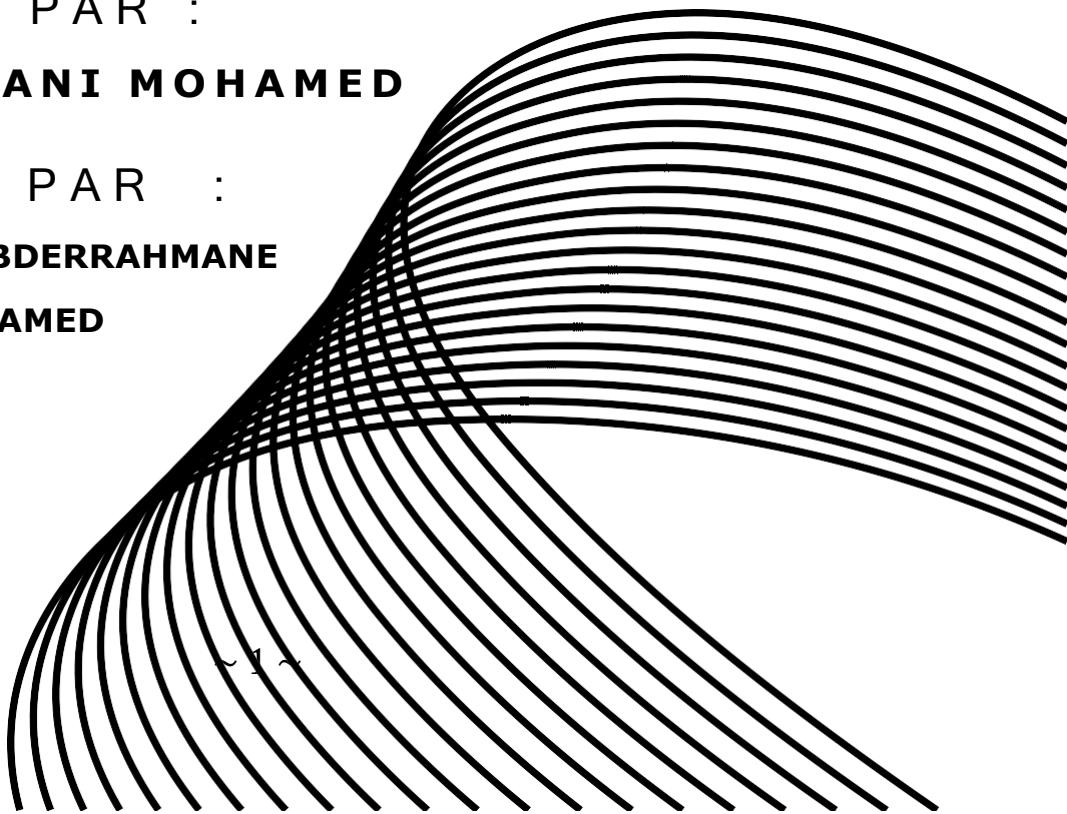
24 / 05 / 2023

RAPPORT TECHNIQUE

LINUX

ENCADRÉ PAR :
MR. GHAILANI MOHAMED

PREPARE PAR :
EL HARRACHY ABDERRAHMANE
ACHERNAN MOHAMED



Sommaire

Postfix :

Etape1-Configuration de hostname	4
Etape2- Installer Postfix	5
Etape3- Configuration Postfix	6
Etape4- Preparation Envoi	8
Etape5- Tester le serveur SMTP	11

Apache :

Etape1-Configuration Avant Installation	12
Etape2- Installer Apache2	12
Etape3- Firewall Configuration	14
Etape4- Création de site Web	16
Etape5-Installing SSL Certificat	19
Etape6- Teste de site web	20
Etape7-Security de site web	20

Nginx :

Etape1- l'installation de "Nginx"	22
Etape2- Gérer le processus "Nginx"	23
Etape3- Fichier et Repertoire:	25
Etape4- Configuration du serveur :	
1. Partie http (Port 80)	29
2. Partie https (Port 443)	30

Apache vs Nginx :

Qu'est-ce que NGINX ?	34
Avantages de NGINX	34
What is Apache?	35
Avantages d'Apache	35
Principales différences entre NGINX et Apache	35
Conversion NGINX , Pourquoi ?	36

DNS :

Qu'est-ce que le DNS (Domain Name System) ?	38
Résolution DNS	38
Configuration DNS dans Notre Projet :	
Etape1- Vérifier la version du serveur :	39
Etape2- Installation Configure DNS (Bind9) service :	40
Etape3- Autorisation d'installation et redémarrage du service bind9.	44
Etape4- Tester service bind9.	45

Docker:

Etape1- l'installation de "Docker"	48
Etape2- Utilisation de la commande Docker.....	51
Etape3- Travailler avec des images Docker	51
Etape4- Dockerizing notre projet:	
a. Configuration du conteneur postgres	55
b. Configuration du conteneur vueJS	58
c. Configuration du conteneur Laravel API	61
d. Configuration du conteneur NGINX	66

Routeur :

Etape1- Connecter à la page de configuration du routeur	69
Etape3- Configuration du routeur	70



Etape1-Configuration de hostname :

d'abord on va commencer par avoir notre hostname :

```
abderrahmane@dns:~$ hostname
dns.hsup.uae.ma
abderrahmane@dns:~$ hostname -f
dns.hsup.uae.ma
abderrahmane@dns:~$ hostname --fqdn
dns.hsup.uae.ma
abderrahmane@dns:~$
```

ou bien on peut faire la commande suivant :

```
$ cd /etc/hostname
```

Aussi on va voir les autres hosts par la commande : [cd /etc/hosts](#)

```
GNU nano 6.2                               /etc/hostname
dns.hsup.uae.ma
```

Etape2- Installer Postfix

Dans cette étape, vous allez installer Postfix. Le moyen le plus rapide est d'installer le paquet mailutils qui regroupe Postfix avec quelques programmes supplémentaires que vous utiliserez pour tester l'envoi de courrier électronique.

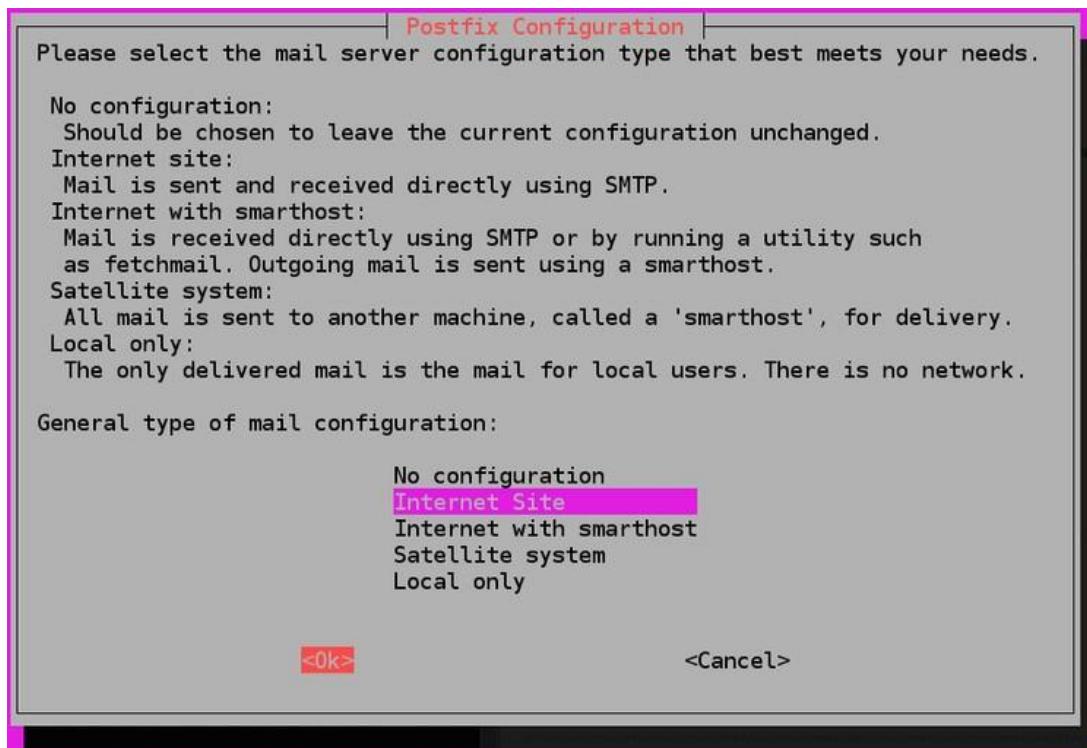
Tout d'abord, mettez à jour la base de données des paquets

```
$ sudo apt update
```

Ensuite, installez Postfix en exécutant la commande suivante :

```
$ sudo apt install mailutils
```

Vers la fin du processus d'installation, la fenêtre de configuration de Postfix vous sera présentée :

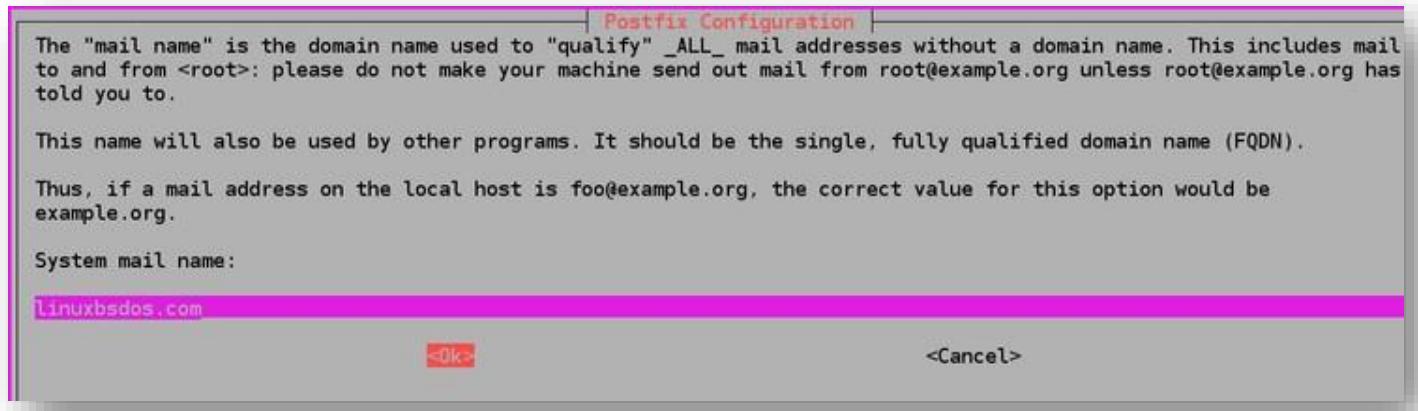


L'option par défaut est `Site Internet`. C'est l'option recommandée pour votre cas d'utilisation, donc appuyez sur `TAB`, puis `ENTRÉE`. Si vous ne voyez que le texte de la description, appuyez sur `TAB` pour sélectionner `OK`, puis sur `ENTER`.

S'il ne s'affiche pas automatiquement, exécutez la commande suivante pour le démarrer :

```
$ sudo dpkg-reconfigure postfix
```

Après cela, vous obtiendrez une autre invite de configuration concernant le **nom de messagerie du système** :



Le **nom de messagerie du système** doit être le même que celui que vous avez attribué à votre serveur lors de sa création. Lorsque vous avez terminé, appuyez sur **TAB**, puis sur **ENTRÉE**.

Vous avez maintenant installé Postfix et vous êtes prêt à commencer à le configurer.

Etape3- Configuration Postfix

Au cours de cette étape, vous configurerez Postfix pour envoyer et recevoir des courriers électroniques uniquement à partir du serveur sur lequel il fonctionne, c'est-à-dire à partir de localhost.

Pour que cela arrive, Postfix doit être configuré pour écouter uniquement sur l'*interface de bouclage*, l'interface de réseau virtuel que le serveur utilise pour communiquer en interne. Pour effectuer les changements, vous devrez modifier le fichier de configuration principal de Postfix appelé main.cf, stocké sous etc/postfix.

Ouvrez-le pour l'éditer à l'aide de votre éditeur de texte préféré :

```
$ sudo nano /etc/postfix/main.cf
```

```

GNU nano 6.2                               /etc/postfix/main.cf
# See /usr/share/postfix/main.cf.dist for a commented, more complete version

# Debian specific: Specifying a file name will cause the first
# line of that file to be used as the name. The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 3.6 on
# fresh installs.
compatibility_level = 3.6

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may

smtp_tls_CPath=/etc/ssl/certs
smtp_tls_security_level=may
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = hsup.vae.ma
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = ${myhostname}, localhost, localhost.localdomain
relayhost =[smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all

```

Ajouter les lignes suivantes :

```

# Enable SASL authentication
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noanonymous
smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd
smtp_tls_security_level = encrypt
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt

```

On va aussi ajouter ce code tout simplement pour personnaliser le format à envoyer

```

#template_email

message_content_type = text/html
html_directory = /usr/share/doc/postfix/html

```

Une autre directive que vous devrez modifier est mydestination, qui est utilisée pour spécifier la liste des domaines qui sont livrés via le transport de distribution de courrier local_transport. Par défaut, les valeurs sont similaires à celles-ci :

```
/etc/postfix/main.cf  
...  
mydestination = $myhostname, your_domain, localhost.com, , localhost  
...
```

Si votre domaine est en fait un sous-domaine, et que vous souhaitez que les messages électroniques aient l'air d'avoir été envoyés depuis le domaine principal, vous pouvez ajouter la ligne suivante à la fin de main.cf:

```
/etc/postfix/main.cf  
...  
masquerade_domains = your_main_domain
```

Le paramètre facultatif masquerade_domains spécifie pour quels domaines la partie de sous-domaine sera supprimée dans l'adresse e-mail.

Lorsque vous avez terminé, enregistrez et fermez le fichier.

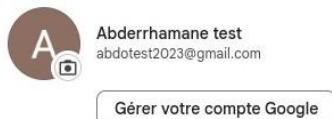
Note : Si vous hébergez plusieurs domaines sur un seul serveur, les autres domaines peuvent également être transmis à Postfix en utilisant la directive mydestination.

Ensuite, redémarrez Postfix en exécutant la commande suivante :

```
$ sudo systemctl restart postfix
```

Etape4- Preparation Envoi

D'abord on doit accéder à l'adresse mail qui va prendre en charge l'opération d'envoi des emails



Google Compte Rechercher dans le compte Google

Accueil Informations personnelles Données et confidentialité Sécurité Contacts et partage Paiements et abonnements À propos

A

Bienvenue Abderrhamane test

Gérez vos informations, ainsi que la confidentialité et la sécurité de vos données pour profiter au mieux des services Google. [En savoir plus](#)

Confidentialité et personnalisation
Gérez les données de votre compte Google et sélectionnez les activités à enregistrer pour personnaliser votre expérience Google.

[Gérer une demande et votre vie privée](#)

Des recommandations de sécurité sont disponibles
Consultez-les sur la page Check-up Sécurité.

[Protéger votre compte](#)

Apres cliquer sur validations en deux étape :

myaccount.google.com/u/1/security?hl=fr

Google Compte Rechercher dans le compte Google

Accueil Informations personnelles Données et confidentialité Sécurité Contacts et partage Paiements et abonnements À propos

Des recommandations de sécurité sont disponibles
Consultez-les sur la page Check-up Sécurité.

[Protéger votre compte](#)

Activité récente liée à la sécurité de votre compte

Mot de passe d'application créé	05:47 - Maroc	>
Mot de passe d'application supprimé	05:47 - Maroc	>
Mot de passe d'application créé	29 avr. - Maroc	>

[Examiner l'activité liée à la sécurité \(19\)](#)

Comment vous connecter à Google
Assurez-vous que vous pouvez toujours accéder à votre compte Google en maintenant ces informations à jour

Validation en deux étapes	Activation : 24 avr.	>
Mot de passe	Dernière modification : 24 avr.	>
Téléphones pour la validation en deux étapes	0626-387972	>
Numéro de téléphone de récupération	0626-387972	>

Click

Apres on click sur Mots de passe des applications

Mots de passe des applications

Les mots de passe d'application ne sont pas recommandés et sont inutiles dans la plupart des cas. Pour sécuriser votre compte Google, utilisez Se connecter avec Google pour y associer des applis.

Mots de passe des applications

1 mot de passe



Apres click sur « sélectionner une application » et choisit « autre »

← Mots de passe des applications

Les mots de passe d'application vous permettent de vous connecter à votre compte Google à partir d'applications sur des appareils non compatibles avec la validation en deux étapes. Comme vous ne devrez saisir le mot de passe qu'une fois, vous n'aerez pas besoin de le mémoriser. [En savoir plus](#)

Nom	Créé le	Dernière utilisation le	
mail	05:47	-	

Sélectionnez l'application et l'appareil pour lesquels vous souhaitez générer le mot de passe d'application.

Sélectionnez une application

- Messagerie**
- Agenda
- Contacts
- YouTube
- Autre (*Nom personnalisé*)

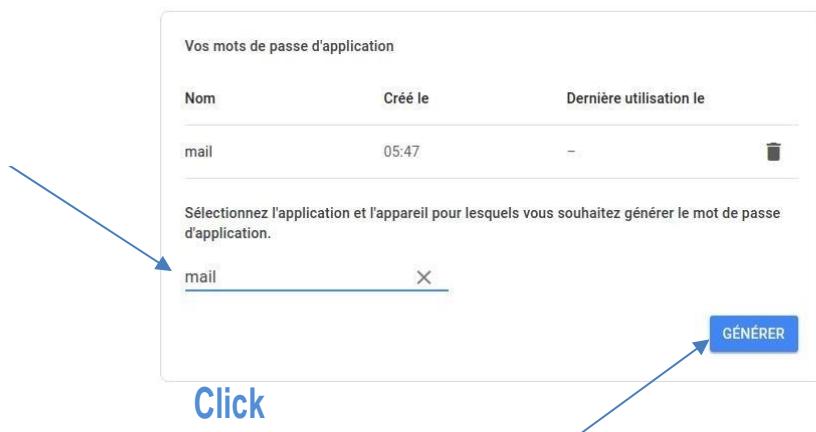
Sélectionnez un appareil

GÉNÉRER

← Mots de passe des applications

Les mots de passe d'application vous permettent de vous connecter à votre compte Google à partir d'applications sur des appareils non compatibles avec la validation en deux étapes. Comme vous ne devez saisir le mot de passe qu'une fois, vous n'avez pas besoin de le mémoriser. En savoir plus

Ecrit par example mail
juste comme Nom



Lorsque vous cliquez sur GENERER
il va donner un mot de passe

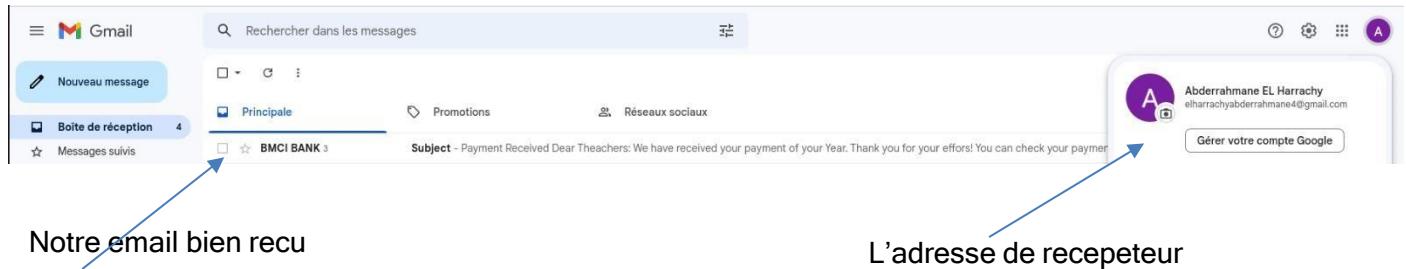
Maintenant on va régler l'émetteur :

```
$ sudo nano /etc/postfix/sasl/sasl_passwd
```

Etape5-Tester le serveur SMTP

```
y
abderrahmane@dns:~$ sudo chown root:root /etc/postfix/sasl/sasl_passwd.db
abderrahmane@dns:~$ sudo chmod 600 /etc/postfix/sasl/sasl_passwd.db
abderrahmane@dns:~$ mail -a "Content-Type: text/html" -s "Subject" elharrachyabderrahmane4@gmail.com
< /usr/share/doc/postfix/html/template.html
```

On verifier dans notre boite mail



Subject: Boîte de réception

BMCI BANK <abdotted2023@gmail.com>
À moi

anglais > français Traduire le message

Payment Received

Dear Teachers:

We have received your payment of your Year. Thank you for your efforts!

You can check your payment details by clicking on the link below:

[View Payment Details](#)

Thank you again for your payment. If you have any questions or concerns, please feel free to contact us.

Best regards,
The Payment Team

Lien de payment de chaque prof



Etape1- Configuration Avant l'installation:

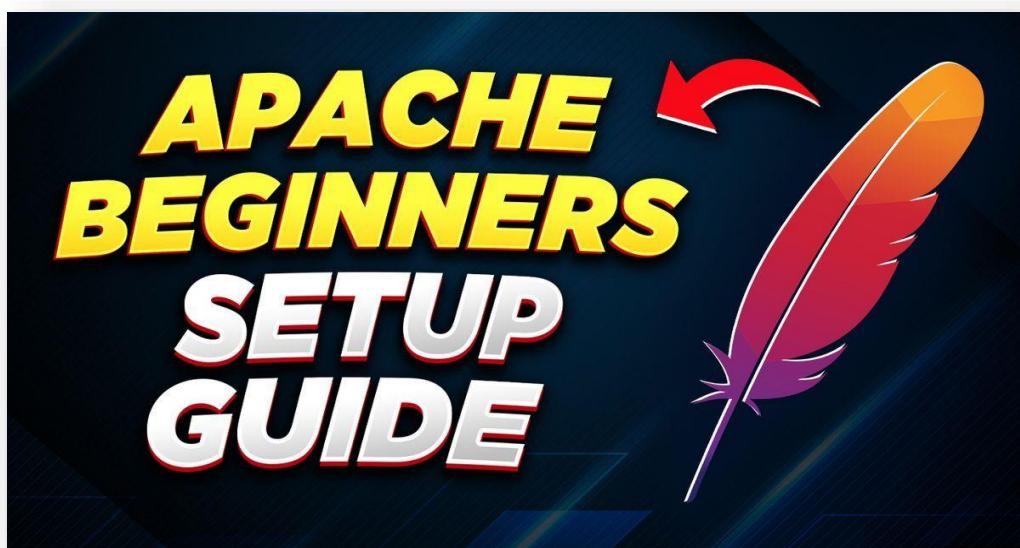
Tout d'abord, mettez à jour la base de données des paquet

```
abderrahmane@dns:~$ sudo apt update && apt upgrade -y
```

Ensuite, modifions nos fichiers hôtes pour qu'ils correspondent au domaine que nous allons héberger.

```
abderrahmane@dns:~$ hostname  
dns.hsup.uae.ma  
abderrahmane@dns:~$ hostname --fqdn  
dns.hsup.uae.ma  
abderrahmane@dns:~$
```

Etape2- Installer Appache2:



Maintenant, nous pouvons installer Apache et configurer notre tout premier site Web. Reconnectez-vous à votre serveur via ssh cette fois en utilisant le nom d'utilisateur au lieu de root. Tapez ensuite la commande ci-dessous pour installer les paquets nécessaires, puis vérifiez s'il est en cours d'exécution.

```
abderrahmane@dns:~$ sudo apt install apache2 apache2-docs apache2-util
```

Maintenant pour avoir le statuts de Apache est ce que sa marche ou Non :

```
$ sudo systemctl status apache2
```

donne quelque chose comme ça :

```
abderrahmane@dns:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-05-17 08:25:31 +01; 6h ago
    Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 928 (apache2)
     Tasks: 55 (limit: 9277)
    Memory: 12.9M
       CPU: 1.951s
      CGroup: /system.slice/apache2.service
              └─ 928 /usr/sbin/apache2 -k start
                  ├─ 8519 /usr/sbin/apache2 -k start
                  ├─ 8520 /usr/sbin/apache2 -k start

May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Starting The Apache HTTP Server...
May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Started The Apache HTTP Server.
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
abderrahmane@dns:~$
```

Active

À ce stade, vous pouvez voir s'il fonctionne sur un navigateur Web et vérifier si le domaine a complètement modifié si vous avez modifié des serveurs de noms. Copiez / collez simplement votre adresse IP dans la barre d'adresse du navigateur Web et vous devriez voir la page par défaut Apache2. Utilisez les commandes ci-dessous pour désactiver le site Web de default car nous créerons nos propres répertoires et configuration plus tard.

Chercher d'abord le fichier 000-default.conf

```
abderrahmane@dns:~$ cd /etc/apache2
abderrahmane@dns:/etc/apache2$ ls
apache2.conf      conf-enabled  magic          mods-enabled  sites-available
conf-available   envvars       mods-available ports.conf    sites-enabled
abderrahmane@dns:/etc/apache2$ cd sites-available
abderrahmane@dns:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  hsup.conf
abderrahmane@dns:/etc/apache2/sites-available$ sudo a2dissite 000-default.conf
```

On redémarre Apache ,et on voir leu status autre fois :

```
$ sudo systemctl reload apache2
```

```
abderrahmane@dns:/etc/apache2/sites-available$ sudo systemctl reload apache2
abderrahmane@dns:/etc/apache2/sites-available$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2023-05-17 08:25:31 +01; 6h ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 12560 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 928 (apache2)
    Tasks: 55 (limit: 9277)
   Memory: 10.5M
      CPU: 2.030s
     CGroup: /system.slice/apache2.service
             └─928 /usr/sbin/apache2 -k start
                  ├─12564 /usr/sbin/apache2 -k start
                  ├─12565 /usr/sbin/apache2 -k start

May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Starting The Apache HTTP Server...
May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Started The Apache HTTP Server.
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
abderrahmane@dns:/etc/apache2/sites-available$
```

Encore active

Etape3- Firewall Configuration:

Maintenant, nous devons activer et autoriser certaines applications avec notre pare-feu. L'application de pare-feu par défaut dans Ubuntu rend cela vraiment facile à faire. Pour voir une liste d'applications, exécutez simplement la commande ci-dessous.

```
$ sudo ufw app list
```

```
abderrahmane@dns:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  Bind9
  CUPS
  OpenSSH
  Postfix
  Postfix SMTPS
  Postfix Submission
abderrahmane@dns:~$
```

Les applications que nous autoriserons sont Apache Full et OpenSSH

```
$ sudo ufw allow 'Apache Full'
$ sudo ufw allow OpenSSH
```

```
abderrahmane@dns:~$ sudo ufw allow 'Apache Full'
Skipping adding existing rule
Skipping adding existing rule (v6)
abderrahmane@dns:~$ sudo ufw allow OpenSSH
Skipping adding existing rule
Skipping adding existing rule (v6)
abderrahmane@dns:~$ sudo ufw enable
Firewall is active and enabled on system startup
abderrahmane@dns:~$
```

De là, activons notre pare-feu et vérifions l'état.

```
$ sudo ufw status
```

```

abderrahmane@dns:~$ sudo ufw status
Status: active

To                         Action      From
--                         ----       ---
Apache                      ALLOW      Anywhere
Apache Full                 ALLOW      Anywhere
22/tcp                      ALLOW      Anywhere
OpenSSH                     ALLOW      Anywhere
20/tcp                      ALLOW      Anywhere
21/tcp                      ALLOW      Anywhere
4100:5100/tcp               ALLOW      Anywhere
990/tcp                     ALLOW      Anywhere
Postfix                     ALLOW      Anywhere
Bind9                       ALLOW      Anywhere
80                          ALLOW      Anywhere
443                         ALLOW      Anywhere
Apache (v6)                  ALLOW      Anywhere (v6)
Apache Full (v6)             ALLOW      Anywhere (v6)
22/tcp (v6)                 ALLOW      Anywhere (v6)
OpenSSH (v6)                 ALLOW      Anywhere (v6)
20/tcp (v6)                  ALLOW      Anywhere (v6)
21/tcp (v6)                  ALLOW      Anywhere (v6)
4100:5100/tcp (v6)          ALLOW      Anywhere (v6)
990/tcp (v6)                 ALLOW      Anywhere (v6)
Postfix (v6)                 ALLOW      Anywhere (v6)
Bind9 (v6)                   ALLOW      Anywhere (v6)
80 (v6)                      ALLOW      Anywhere (v6)
443 (v6)                     ALLOW      Anywhere (v6)

abderrahmane@dns:~$
```

Etape4- Creating Website:

La création de répertoires pour notre site Web est facile et se fera dans le répertoire

```

abderrahmane@dns:~$ cd /var/www
abderrahmane@dns:/var/www$ ls
hsup  hsup1
abderrahmane@dns:/var/www$
```

/var/www/html par défaut. Avec les commandes ci-dessous, remplacez 'example.com' par le domaine que vous utiliserez.

On va créer notre nouveau site Web

```

abderrahmane@dns:/var/www$ cd hsup
abderrahmane@dns:/var/www/hsup$ ls
css  favicon.ico  index.html  js
abderrahmane@dns:/var/www/hsup$
```

Remarque : que sont des fichiers de notre test web quand 'on créer avec vuejs .

C'est le path de dossier que nous avons faire

```
abderrahmane@dns: $ cd ~/App
abderrahmane@dns:~/App$ ls
Linux-Server-docker-test
abderrahmane@dns:~/App$ cd Linux-Server-docker-test
abderrahmane@dns:~/App/Linux-Server-docker-test$ ls
api app docker-compose.yml README.md
babel.config.js dist Dockerfile jsconfig.json node_modules package.json package-lock.json postcss.config.js public README.md src tailwind.config.js vue.config.js
css favicon.ico index.html js
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$ ls
```

On copie le fichier de deployment (dist) dans notre dossier (site Web).

```
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$ ls
css favicon.ico index.html js
abderrahmane@dns:~/App/Linux-Server-docker-test/app/dist$ cd ..
abderrahmane@dns:~/App/Linux-Server-docker-test/app$ sudo cp dist /var/www/hsup
```

Fichier deployment Site Web

Une fois nos répertoires créés, nous allons revenir à l'emplacement des configurations de site et en créer un nouveau pour notre site Web.

```
$ cd /etc/apache2/sites-available
```

```
abderrahmane@dns:/etc/apache2/sites-available$ ls
000-default.conf default-ssl.conf hsup.conf
abderrahmane@dns:/etc/apache2/sites-available$ sudo nano hsup.conf
abderrahmane@dns:/etc/apache2/sites-available$
```

Dans cette configuration, vous voudrez copier / coller la configuration ci-dessous.
Remplacer tout ce qui correspond à votre configuration.

```
GNU nano 6.2                                     hsup.conf
<VirtualHost *:80>
    ServerName hsup.uae.ma
    Redirect / https://hsup.uae.ma/               ← La redirection vers HTTPS

<Directory /var/www/hsup>
    Options -Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

</VirtualHost>

<VirtualHost *:443>
    ServerName hsup.uae.ma
    ServerAlias www.hsup.uae.ma
    DocumentRoot /var/www/hsup

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

Concernant la certificate SSL

Maintenant, nous exécutons quelques commandes pour activer notre nouvelle configuration.

```
abderrahmane@dns:/etc/apache2/sites-available$ sudo a2ensite hsup
Site hsup already enabled
abderrahmane@dns:/etc/apache2/sites-available$ █
```

```

abderrahmane@dns:/etc/apache2/sites-available$ systemctl reload apache2
abderrahmane@dns:/etc/apache2/sites-available$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-05-17 08:25:31 +01; 6h ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 13074 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 928 (apache2)
    Tasks: 55 (limit: 9277)
   Memory: 10.5M
      CPU: 2.149s
      CGroup: /system.slice/apache2.service
              └─ 928 /usr/sbin/apache2 -k start
                  ├─13078 /usr/sbin/apache2 -k start
                  ├─13079 /usr/sbin/apache2 -k start

May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Starting The Apache HTTP Server...
May 17 08:25:31 dns.hsup.uae.ma systemd[1]: Started The Apache HTTP Server.
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:48:14 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 12:52:20 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 14:53:07 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
May 17 15:06:59 dns.hsup.uae.ma systemd[1]: Reloading The Apache HTTP Server...
May 17 15:06:59 dns.hsup.uae.ma systemd[1]: Reloaded The Apache HTTP Server.
abderrahmane@dns:/etc/apache2/sites-available$
```

Maintenant, si vous voyagez sur votre site Web, vous devriez voir la page standard 'Index de /' signifiant jusqu'à présent, nous avons réussi! De là, vous pouvez créer un fichier index.html pour confirmer à coup sûr. Voir l'exemple dans la vidéo de la façon de procéder.

Etape5- Installing SSL Certification

Un certificate SSL est un certificate numérique qui authentifie l'identité d'un site Web et permet une connexion cryptée. SSL signifie Secure Sockets Layer, un protocole de sécurité qui crée un lien crypté entre un serveur Web et un navigateur Web. Dans le Web moderne, c'est une exigence de base. Pour installer un certificat SSL, assurez-vous que votre domaine est correctement lié à votre nouveau serveur Apache et suivez les étapes ci-dessous.

```
$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Apres ca il faut remplire les informations .

Etape6- Teste de Site Web



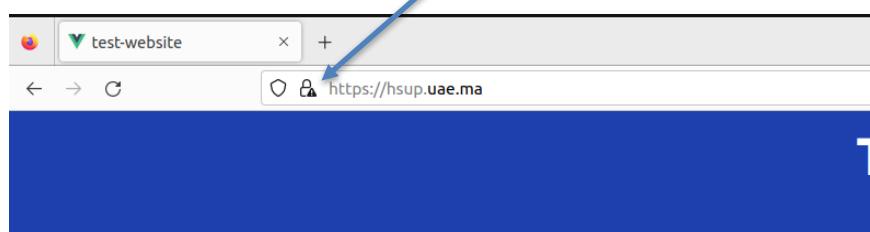
On Remarque il fait la direction directement vers https de plus il est en relation avec Vuejs.



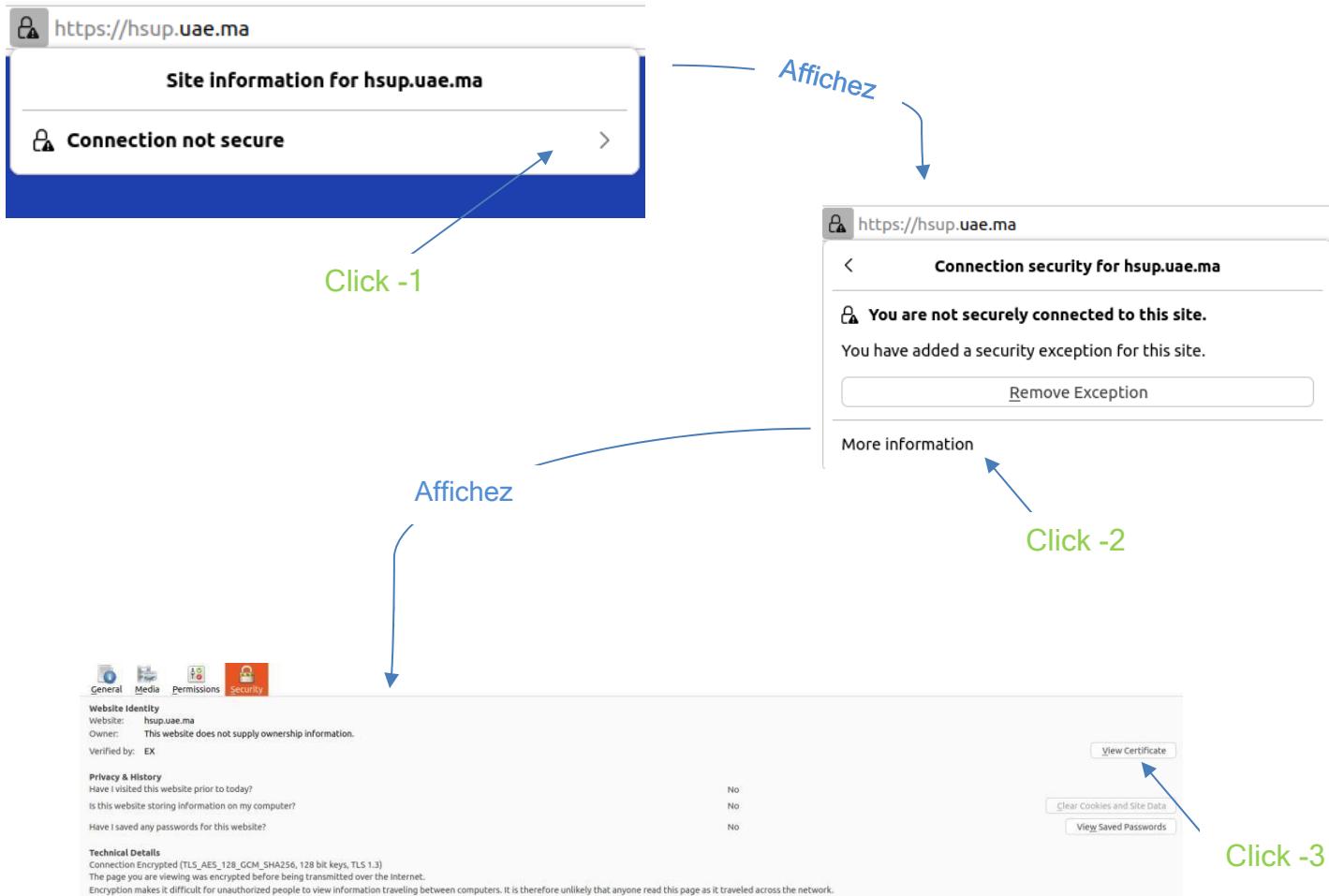
This is just a test website. Don't expect anything.

Etape7- Security de Site Web:

On Remarque que on un signe Serrure,Si on Click



Il va afficher l'icône suivant :



Et voilà notre certificate

Certificate

hsup.uae.ma

Subject Name

Country	MR
State/Province	Some-State
Organization	Internet Widgits Pty Ltd
Common Name	hsup.uae.ma
Email Address	abdotest2023@gmail.com

Issuer Name

Country	MR
State/Province	Some-State
Organization	Internet Widgits Pty Ltd
Common Name	hsup.uae.ma
Email Address	abdotest2023@gmail.com

Validity

Not Before	Wed, 17 May 2023 14:15:25 GMT
Not After	Thu, 16 May 2024 14:15:25 GMT

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	BC:4D:B5:A9:52:37:B6:14:70:10:43:6B:58:D9:8C:34:3D:E6:45:35:32:90:5A:E1...

Cependant, si vous utilisez un certificat auto-signé, il n'est pas signé par une autorité de certification reconnue. Par conséquent, les navigateurs et les systèmes d'exploitation considéreront ce certificat comme non fiable par défaut. Lorsque vous accédez à un site web avec un certificat auto-signé, vous recevrez généralement un avertissement de sécurité indiquant que le certificat n'est pas valide.

Cela signifie que bien que votre serveur web ait un certificat SSL installé, il sera considéré comme non fiable par les navigateurs des utilisateurs, ce qui entraînera des avertissements de sécurité.



Étape 1 : l'installation de "Nginx":

Nginx est disponible dans les référentiels par défaut d'Ubuntu, l'installation est donc assez simple.

système de **packaging apt** dans cette session. Nous mettrions à jour notre index de packages local ; afin que nous ayons accès aux listes de packages les plus récentes.

Ensuite, nous pouvons installer Nginx en utilisant la commande suivante



Bash Terminal (Ubuntu)

```
> sudo apt-get update  
> sudo apt-get install nginx
```

Après avoir accepté la procédure, apt-get installera **Nginx** et toutes les dépendances requises sur votre serveur.

Étape 2 : Gérer le processus "Nginx":

Maintenant que le serveur Web est opérationnel, nous pouvons utiliser certaines commandes de gestion de base pour contrôler l'état du serveur.

1. Pour arrêter votre serveur Web:



Bash Terminal (Ubuntu)

```
> sudo systemctl stop nginx
```

The terminal window shows the execution of the command `sudo systemctl stop nginx`. The output indicates that the Nginx service is inactive (dead) since the previous day. A yellow arrow points to the status line: ***le serveur est inactive**.

```
sudo systemctl stop nginx
*le serveur est inactive
sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Sun 2023-05-21 20:34:37 WEST; 2s ago
    Docs: man:nginx(8)
   Process: 3929 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/n
 Main PID: 879 (code=exited, status=0/SUCCESS)
   CPU: 61ms

mai 21 19:55:01 mohamed-X541UJ systemd[1]: Starting A high performance web server and a reverse p>
mai 21 19:55:01 mohamed-X541UJ nginx[872]: nginx: [warn] the "ssl" directive is deprecated, use t>
mai 21 19:55:01 mohamed-X541UJ nginx[878]: nginx: [warn] the "ssl" directive is deprecated, use t>
mai 21 19:55:01 mohamed-X541UJ systemd[1]: Started A high performance web server and a reverse pr>
mai 21 20:34:37 mohamed-X541UJ systemd[1]: Stopping A high performance web server and a reverse p>
mai 21 20:34:37 mohamed-X541UJ systemd[1]: nginx.service: Deactivated successfully.
mai 21 20:34:37 mohamed-X541UJ systemd[1]: Stopped A high performance web server and a reverse pr>
```

2. Pour démarrer le serveur Web lorsqu'il est arrêté:



Bash Terminal (Ubuntu)

```
> sudo systemctl start nginx
```

```
INT X 3m 12s ✘
[sudo] * ~
  sudo systemctl start nginx
[sudo] * ~
  sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2023-05-21 20:37:55 WEST; 2s ago
    Docs: man:nginx(8)
   Process: 3971 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0)
   Process: 3972 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0)
 Main PID: 3973 (nginx)
    Tasks: 5 (limit: 9307)
   Memory: 5.5M
      CPU: 40ms
     CGroup: /system.slice/nginx.service
```

*Le serveur est maintenant actif et en cours d'exécution ✓

Si vous apportez simplement des modifications de configuration, NGINX peut souvent recharger sans interrompre les connexions. Pour ce faire, cette commande peut être utilisée



Bash Terminal (Ubuntu)

> sudo systemctl reload nginx // refrecher le serveur

Pour arrêter puis remettre le service



Bash Terminal (Ubuntu)

> sudo systemctl restart nginx // redémarrer le serveur

Par défaut, NGINX est configuré pour démarrer automatiquement au démarrage du serveur. Si ce n'est pas ce que vous voulez, vous pouvez désactiver ce comportement en tapant :



Bash Terminal (Ubuntu)

> sudo systemctl disable nginx // désactiver le serveur

Pour réactiver le service afin qu'il démarre au démarrage



Bash Terminal (Ubuntu)

```
> sudo systemctl enable nginx // ré-activer le serveur
```



Bash Terminal (Ubuntu)

```
> sudo nginx -t // tester le fichier conf
```

```
sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

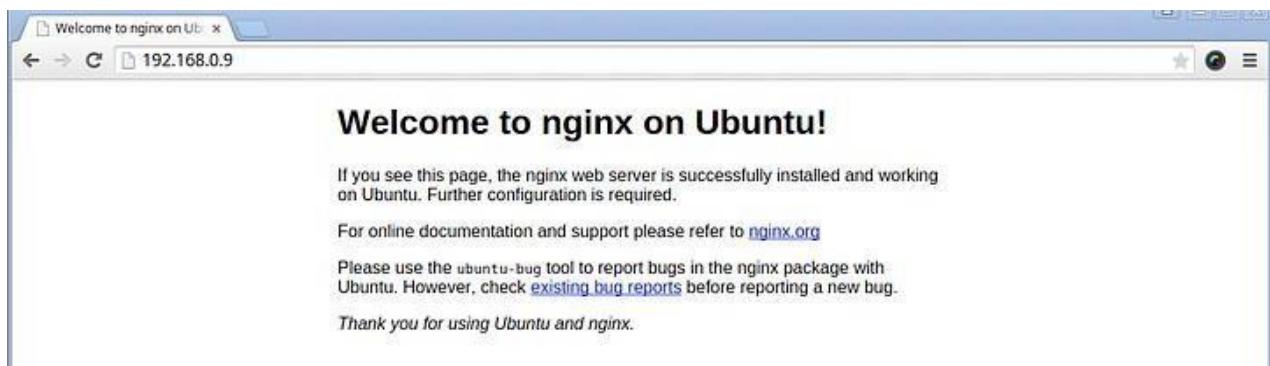
Dans ce cas "hsup-nginx" est le fichier de configuration utilisé pour configurer le serveur nginx pour nos sites Web et le test indique qu'il n'y a pas de syntaxe ni d'erreur de configuration

Étape 3 : Fichier et Repertoire:

1. Contenu:

</var/www/html> : le contenu Web réel, qui par défaut se compose uniquement de la page Nginx par défaut illustrée dans la capture d'écran, est diffusé à partir du répertoire
/var/www/html.

Cela peut être modifié en modifiant les fichiers de configuration NGINX.



2. Configuration du serveur:

[/etc/nginx](#) : le répertoire de configuration de NGINX. Tous les fichiers de configuration Nginx résident ici.

```
ll /etc/nginx
total 64K
drwxr-xr-x 2 root root 4,0K nov. 10 2022 conf.d
-rw-r--r-- 1 root root 1,1K juil. 27 2022 fastcgi.conf
-rw-r--r-- 1 root root 1,1K juil. 27 2022 fastcgi_params
-rw-r--r-- 1 root root 2,8K juil. 27 2022 koi-utf
-rw-r--r-- 1 root root 2,2K juil. 27 2022 koi-win
-rw-r--r-- 1 root root 3,9K août 2 2022 mime.types
drwxr-xr-x 2 root root 4,0K nov. 10 2022 modules-available
drwxr-xr-x 2 root root 4,0K mai 18 23:09 modules-enabled
-rw-r--r-- 1 root root 1,5K juil. 27 2022 nginx.conf
-rw-r--r-- 1 root root 180 juil. 27 2022 proxy_params
-rw-r--r-- 1 root root 636 juil. 27 2022 scgi_params
drwxr-xr-x 2 root root 4,0K mai 18 23:09 sites-available
drwxr-xr-x 2 root root 4,0K mai 21 21:15 sites-enabled
drwxr-xr-x 2 root root 4,0K mai 18 23:09 snippets
-rw-r--r-- 1 root root 664 juil. 27 2022 uwsgi_params
-rw-r--r-- 1 root root 3,0K juil. 27 2022 win-utf
```

*Tous les fichiers reliés à la configuration du serveur NGINX résident dans cette répertoire

[/etc/nginx/nginx.conf](#) : fichier de configuration principal de NGINX. Cela peut être modifié pour apporter des modifications à la configuration globale de NGINX.

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;
}

# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/xml application/xml+rss text/javascript;

## 
# Virtual Host Configs
## 

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
```

Ce qui est très important, c'est de s'assurer que les fichiers de configuration appropriés tels que les fichiers .conf et les fichiers du dossier sites-enabled sont inclus comme indiqués dans la capture d'écran.

/etc/nginx/sites-available/ : le répertoire dans lequel les "blocs de serveur" par site peuvent être stockés. NGINX n'utilisera pas les fichiers de configuration trouvés dans ce répertoire au moins qu'ils ne soient liés au répertoire sites-enabled (voir ci-dessous). En règle générale, toute la configuration du bloc de serveur est effectuée dans ce répertoire, puis activée en établissant un lien vers l'autre répertoire.

/etc/nginx/snippets : ce répertoire contient des fragments de configuration qui peuvent être inclus ailleurs dans la configuration NGINX. Les segments de configuration potentiellement reproductibles sont de bons candidats pour la refactorisation en extraits.

/etc/nginx/sites-enabled/ : le répertoire dans lequel les "blocs de serveur" activés par sites sont stockés. En règle générale, ceux-ci sont créés en payant un lien vers les fichiers de configuration trouvés dans le répertoire des sites disponibles.

```
ls -lai
total 12
820043 drwxr-xr-x 2 root root 4096 mai 21 21:15 .
819905 drwxr-xr-x 8 root root 4096 mai 20 23:37 ..
820104 lrwxrwxrwx 1 root root 34 mai 18 23:09 default -> /etc/nginx/sites-available/default
788990 -rw-r--r-- 1 root root 623 mai 21 21:15 hsup-nginx
```

*Le fichier de conf par défaut

*Le fichier de configuration de notre serveur

Journaux du serveur

/var/log/nginx/access.log : chaque demande envoyée à notre serveur Web est enregistrée dans ce fichier journal, au moins et jusqu'à ce que NGINX soit configuré pour faire autrement.

/var/log/nginx/error.log : chaque erreur NGINX sera enregistrée dans ce journal uniquement.

Étape 4 : Configuration du serveur:

Nous devons configurer NGINX pour servir notre site Web.

cd dans /etc/nginx/. C'est là que se trouvent les fichiers de configuration NGINX. Les deux répertoires qui nous intéressent sont les sites disponibles et les sites activés.

sites-available : contient des fichiers de configuration individuels pour tous vos sites Web statiques possibles.

sites-enabled : des liens vers les fichiers de configuration que NGINX va réellement lire et contenir.

Maintenant, **cd dans /etc/nginx/sites-enabled** et éditez le fichier par défaut en utilisant neo-vim ou nano selon votre choix comme suit :

1. Partie http (Port 80):

Dans ce cas, nous avons choisi de servir le site Web à l'adresse "**192.168.1.109**"

```
server {
    listen 80;
    listen 192.168.1.109:80;
    server_name www.hsup.uae.ma hsup.uae.ma;
    return 301 https://$server_name$request_uri;

    root /vue-app/dist;
    index index.html index.htm;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

- **listen 80** : Le port 80 est le numéro de port attribué au protocole de communication Internet couramment utilisé, le protocole de transfert hypertexte (HTTP). Il s'agit du port réseau par défaut utilisé pour envoyer et recevoir des pages Web non chiffrées.

- **listen 192.168.1.109:80** : Le serveur doit écouter les requêtes qui passent par cette adresse IP "192.168.1.109" et elles doivent être servies par ce bloc de serveur.

- **server_name www.hsup.uae.ma hsup.uae.ma** : Le nom du serveur qui doit être configuré dans /etc/hosts afin que tout le trafic et les requêtes passant par "**192.168.1.109:80**" soient redirigés vers "www. hsup.uae.ma" et/ou "**hsup.uae.ma**"

```

127.0.0.1      localhost
127.0.1.1      mohamed-X541UJ
# 0.0.0.0        hsup.uae.ma
# 0.0.0.0        www.hsup.uae.ma

# Local Network Acess:
192.168.1.109  hsup.uae.ma
192.168.1.109  www.hsup.uae.ma

# 192.168.1.151 hsup.uae.ma
# 192.168.1.151 www.hsup.uae.ma

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
/etc/hosts

```

1,1 All

*rediriger tous les requêtes et le traffic qui vient à travers "192.168.1.109" vers les adresses: "hsup.uae.ma" et "www.hsup.uae.ma"

root /app/dist : Le répertoire racine où l'on peut trouver le dossier de déploiement du projet vuejs.

renvoie 301 https://\$server_name\$request_uri ; : rediriger tout le trafic provenant du site http vers le site https

```

location / {
try_files $uri $uri/ /index.html;
}

```

Utilisez le fichier index.html et utilisez-le comme **FallBackAddress** vers d'autres **routes** et **points de terminaison**

2. Partie https (Port 443):

```
server {
    listen 443 ssl;
    listen 192.168.1.109:443;
    server_name www.hsup.uae.ma hsup.uae.ma;

    root /vue-app/dist;
    index index.html index.htm;

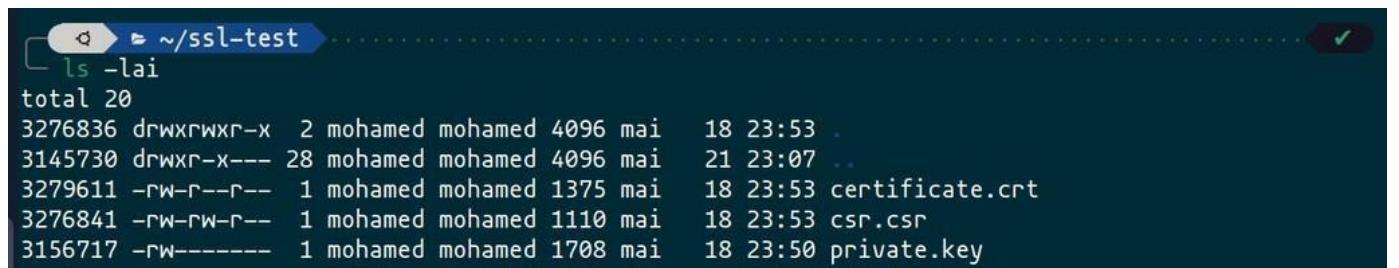
    ssl_certificate /home/mohamed/ssl-test/certificate.crt;
    ssl_certificate_key /home/mohamed/ssl-test/private.key;
    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

listen 443 ssl: : Port 443. L'Internet Engineering Task Force (IETF) reconnaît le numéro de port TCP 443 comme protocole HTTPS par défaut. Il fournit un algorithme de cryptage pour l'échange d'informations entre les serveurs Web et les navigateurs. Le port HTTPS 443 fonctionne en sécurisant les paquets de trafic réseau avant que la transmission de données ne se produise.

SSL (Secure Sockets Layer) et son successeur, **TLS** (Transport Layer Security), sont des protocoles permettant d'établir des liens authentifiés et cryptés entre des ordinateurs en réseau.

ssl_certificate /home/mohamed/ssl-test/certificate.crt;
ssl_certificate_key /home/mohamed/ssl-test/private.key ;

L'emplacement du fichier pour le certificat ssl auto-signé et la clé de certificat ssl. Ce certificat est auto-signé et indique au navigateur "**NON SÉCURISÉ**" lors de l'utilisation du site Web. Pour plus de détails, consultez la partie Configuration SSL



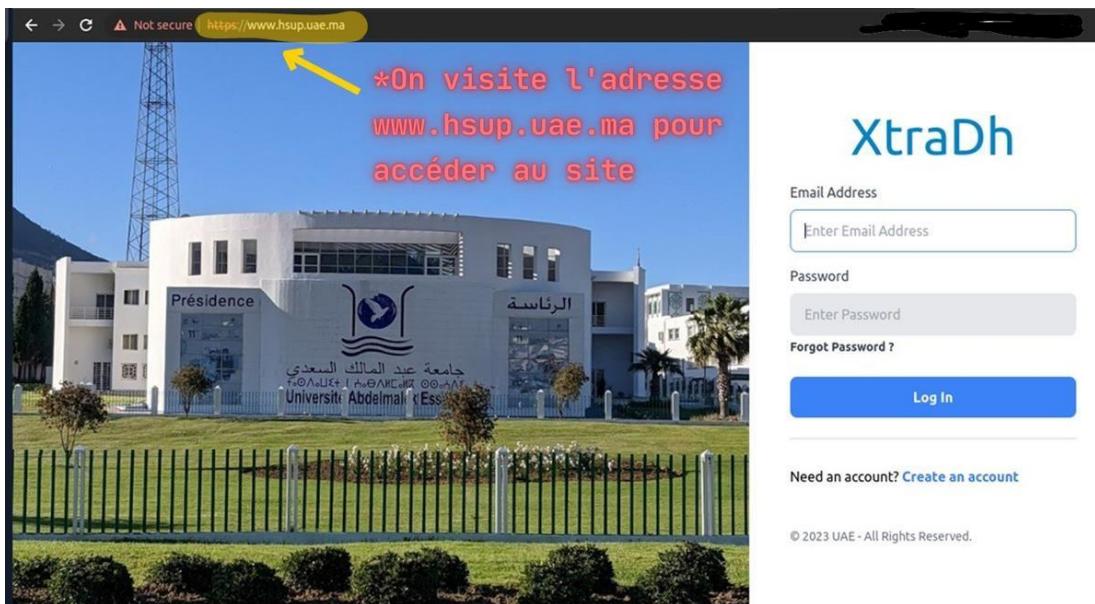
```
ls -lai
total 20
3276836 drwxrwxr-x  2 mohamed mohamed 4096 mai 18 23:53 .
3145730 drwxr-x--- 28 mohamed mohamed 4096 mai 21 23:07 ..
3279611 -rw-r--r--  1 mohamed mohamed 1375 mai 18 23:53 certificate.crt
3276841 -rw-rw-r--  1 mohamed mohamed 1110 mai 18 23:53 csr.csr
3156717 -rw-------  1 mohamed mohamed 1708 mai 18 23:50 private.key
```

Tous les autres paramètres sont les mêmes que la partie HTTP

Après avoir terminé la configuration, nous pouvons démarrer le serveur afin de tester les fonctionnalités que nous avons configurées précédemment

```
ls -l ~ssl-test
[sudo] password for mohamed:
● nginx.service - A high performance web server and a reverse proxy server
  Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2023-05-21 20:37:55 WEST; 2h 31min ago
    Docs: man:nginx(8)
Process: 3971 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0)
Process: 3972 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0)
Main PID: 3973 (nginx)
      Tasks: 5 (limit: 9307)
     Memory: 7.0M
        CPU: 72ms
       CGroup: /system.slice/nginx.service
           ├─3973 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           ├─3974 "nginx: worker process"    ...
           ├─3975 "nginx: worker process"    ...
           ├─3976 "nginx: worker process"    ...
           └─3977 "nginx: worker process"    ...
```

On le test sur les navigateur (Le site Web présenté dans ce rapport est différent de la partie apache2 car de nombreuses mises à jour ont été effectuées au cours de la période ultérieure.)



Nous utilisons l'onglet réseau du navigateur Chrome/Mozilla pour surveiller le trafic du site Web

The screenshot shows the Network tab in the Chrome DevTools. A request to `www.hsup.uae.ma` is selected. The 'General' section of the right panel displays the following details:

- Request URL: `https://www.hsup.uae.ma/`
- Request Method: GET
- Status Code: 200 OK
- Remote Address: 192.168.1.109:443 (highlighted with a yellow box and arrow)
- Referrer Policy: strict-origin-when-cross-origin

Below the general details, the 'Response Headers' section lists:

- Connection: keep-alive
- Content-Encoding: gzip
- Content-Type: text/html
- Date: Sun, 21 May 2023 22:16:37 GMT
- ETag: W/"646a0528-25e"
- Last-Modified: Sun, 21 May 2023 11:48:56 GMT
- Server: nginx/1.18.0 (Ubuntu)
- Transfer-Encoding: chunked

On the left side of the screenshot, there is a redacted login form for 'XtraDh' with fields for Email Address, Password, and Log In button.

***On voit
l'adresse
et le
port
utilisé**

Si nous essayons d'accéder à "<http://www.hsup.uae.ma>", le serveur lèvera un code d'état 301 et redirigera la requête vers l'adresse **HTTPS**

The screenshot shows the Network tab in the Chrome DevTools. A request to `http://www.hsup.uae.ma/` is selected. The 'General' section of the right panel displays the following details:

- Request URL: `http://www.hsup.uae.ma/`
- Request Method: GET
- Status Code: 301 Moved Permanently (highlighted with a yellow box and arrow)
- Remote Address: 192.168.1.109:80
- Referrer Policy: strict-origin-when-cross-origin

Below the general details, the 'Response Headers' section lists:

- Connection: keep-alive
- Content-Length: 178
- Content-Type: text/html
- Date: Sun, 21 May 2023 22:19:48 GMT
- Location: `https://www.hsup.uae.ma/`
- Server: nginx/1.18.0 (Ubuntu)

On the left side of the screenshot, there is a redacted login form for 'XtraDh' with fields for Email Address, Password, and Log In button.

***rediriger
vers HTTPS**

La même chose s'est adaptée si l'utilisateur a essayé d'accéder à l'adresse IP du serveur

Dimensions: Responsive ▾ 598 x 581 100% ▾ No throttling ▾

XtraDh

Email Address
Enter Email Address

Password
Enter Password

Forgot Password ?

Log In

*rediriger vers HTTPS

Need an account? [Create an account](#)

© 2023 UAE - All Rights Reserved.

Network tab details:

- Request URL: http://192.168.1.109/
- Request Method: GET
- Status Code: 301 Moved Permanently (from disk cache)
- Remote Address: 192.168.1.109:80
- Referrer Policy: strict-origin-when-cross-origin

Response Headers:

- Content-Length: 178
- Content-Type: text/html
- Date: Sun, 21 May 2023 21:57:51 GMT
- Location: https://www.hsup.uae.ma/
- Server: nginx/1.18.0 (Ubuntu)

Nous remarquons que toutes les routes principales et secondaires fonctionnent également correctement

Not secure | https://www.hsup.uae.ma/Gestionp

Admin Etab

- Gestion des profs
- + Gestion des Interventions

***le sous-route www.hsup.uae.ma/Gestionp fonctionne**

Listes des profs

PPR	NOM	PRENOM	DATE_NAISSANCE	ETABLISSEMENT

Ajouter prof

Not secure | https://www.hsup.uae.ma/Gestqhqh

Welcome to your home

*Une adresse aléatoire

404 | Not found

[BACK TO LOGIN](#)



Qu'est-ce que NGINX ?

NGINX (également écrit Nginx) est un logiciel open source utilisé pour le service Web, le proxy inverse, la mise en cache, l'équilibrage de charge, le streaming multimédia, etc. NGINX est entré en scène en 2004, lorsqu'il a été rendu public pour la première fois par le développeur russe Igor Sysoev. Lors de sa sortie, NGINX était principalement utilisé pour servir des fichiers statiques, mais aujourd'hui, il est devenu un serveur Web complet qui traite l'ensemble des tâches de serveur.

Avantages de NGINX :

Le serveur **NGINX** présente de nombreux avantages. Certains d'entre eux sont les suivants :

- **NGINX** est open source et son utilisation est gratuite.
- **NGINX** a un système de configuration simple, contrairement à Apache. L'utilisateur ne doit pas modifier plus de deux fichiers.
- **Les performances de NGINX** lorsqu'il s'agit de servir des fichiers statiques tels que pdf, zip, HTML, mp4, MPEG, Avi-, jpg, gif, png et autres sont excellentes.

- **NGINX** gère le trafic de n'importe quelle application Web sans aucune difficulté et fournit des guides de configuration pour la plupart des applications. Diverses applications telles que WordPress, python, ruby, Drupal, VBulletin, PhpBB et de nombreuses autres applications populaires sont servies par NGINX.

- **NGINX** est supérieur en matière de gestion des connexions simultanées, de temps de réponse et d'utilisation des ressources par rapport à d'autres serveurs Web tels qu'Apache ou Lighttpd.

- **NGINX** rend les sites Web plus rapides, leur permettant ainsi d'obtenir un classement plus élevé par Google, car ils ont déjà apporté diverses modifications à l'algorithme de classement de recherche et, récemment, ils ont également intégré la vitesse du site Web dans l'algorithme. Les sites Web plus rapides bénéficient d'une assistance maximale de Google. NGINX peut gérer plusieurs demandes de clients simultanément et efficacement avec un trafic important. Cela rend le site Web plus optimisé et plus rapide.

What is Apache?

- Apache (également connu sous le nom de Apache HTTP Server ou httpd) est un serveur Web multiplateforme open source qui fournit du contenu Web via Internet. Il a été conçu par Tim Berners Lee et publié en 1995. Apache est rapidement devenu populaire et est maintenant maintenu par l'Apache Software Foundation. Comme tous les serveurs Web, Apache héberge du contenu Web, tel que des pages HTML, des fichiers PHP, des fichiers audio et vidéo, et fournit le contenu aux utilisateurs lorsqu'ils visitent un site Web.

- Apache est le premier choix parmi les administrateurs de serveur en raison de sa simplicité architecturale, de sa flexibilité, de sa compatibilité de puissance et de sa prise en charge multiplateforme. Il est compatible avec presque tous les systèmes d'exploitation tels que Windows, UNIX, OSX, NetWare, etc. Mais il est couramment utilisé en combinaison avec Linux.

Avantages d'Apache:

- **Apache** est open source, ce qui le rend gratuit et permet aux développeurs de la communauté de l'étendre et de le modifier facilement.

- **Apache** est flexible grâce à la présence de modules dynamiques.

- **Apache** est compatible avec les systèmes d'exploitation de type Unix tels que Linux, BSD, etc. et MS Windows.

- **Apache** a une énorme communauté et un support facilement disponible en cas de problème.

- **Apache** a une délivrabilité optimale pour les fichiers statiques et est compatible avec n'importe quel langage de programmation.

Principales différences entre NGINX et Apache :

• **NGINX** est un serveur Web open source hautes performances et un serveur proxy inverse, tandis qu'**Apache** est un serveur HTTP open source. **NGINX** étant un serveur proxy inverse, il se trouve derrière le pare-feu dans un réseau privé et dirige les demandes des clients vers le serveur principal approprié, assurant un flux fluide du trafic réseau entre les clients et les serveurs.

• La différence fondamentale entre les deux réside dans leur architecture. **NGINX** a une architecture à thread unique dans laquelle plusieurs requêtes client sont traitées dans un seul thread. D'autre part, **Apache** a une architecture multi-thread où un processus est créé pour une requête. L'architecture événementielle de **NGINX** garantit de meilleures performances même en cas de trafic intense.

• **NGINX** est bien équipé pour gérer le contenu statique. C'est 2,5 fois plus rapide qu'**Apache**. En ce qui concerne le contenu dynamique, **Apache** traite le contenu dynamique de manière native au sein du serveur Web lui-même, tandis que **NGINX** ne peut pas traiter le contenu dynamique en interne et dépend de processus externes pour l'exécution.

• Les serveurs **NGINX** et **Apache** sont tous deux excellents pour gérer la sécurité, mais la base de code **NGINX** est nettement plus petite, c'est donc un gros plus du point de vue de la sécurité.

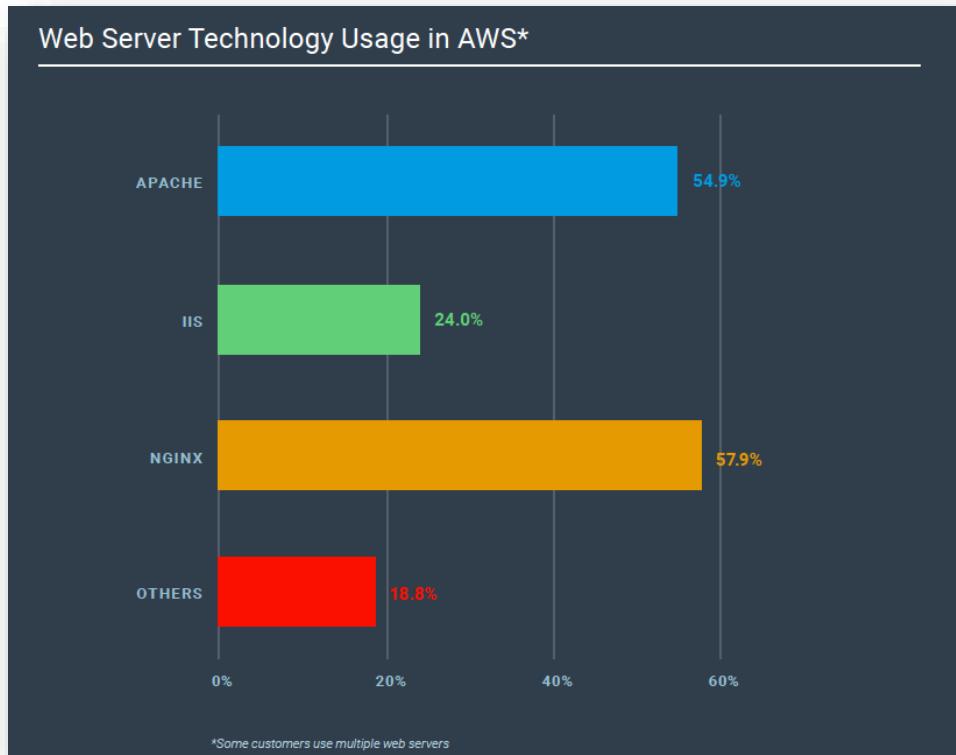
• **NGINX** prend entièrement en charge presque tous les systèmes d'exploitation Unix, mais la prise en charge de Windows est partielle. D'autre part, **Apache** prend entièrement en charge les systèmes d'exploitation de type Unix et MS Windows.

Conversion NGINX , Pourquoi ?

Il est juste de dire que NGINX et Apache offrent des performances de qualité - ils sont flexibles, capables et puissants. Les deux ont leur place sur le marché actuel.

Mais NGINX est celui qui a le vent en poupe. Lorsqu'il était nouveau, Apache a capturé l'espace du serveur Web, mais il y a eu progressivement un changement. NGINX est léger et possède une architecture événementielle. Grâce à quoi il peut gérer plus efficacement un trafic Web important. De plus, il est plus rapide que le traitement de fichiers statiques. Ces

fonctionnalités ont plus de valeur aujourd'hui où le streaming et la consommation d'Internet sont à leur apogée. Par conséquent, les développeurs trouvent l'architecture NGINX plus adaptée au développement d'applications Web modernes. Cela est évident par le graphique ci-dessous montrant l'utilisation des serveurs Web sélectionnés pour les sites Web





Qu'est-ce que le *DNS* (Domain Name System) ?

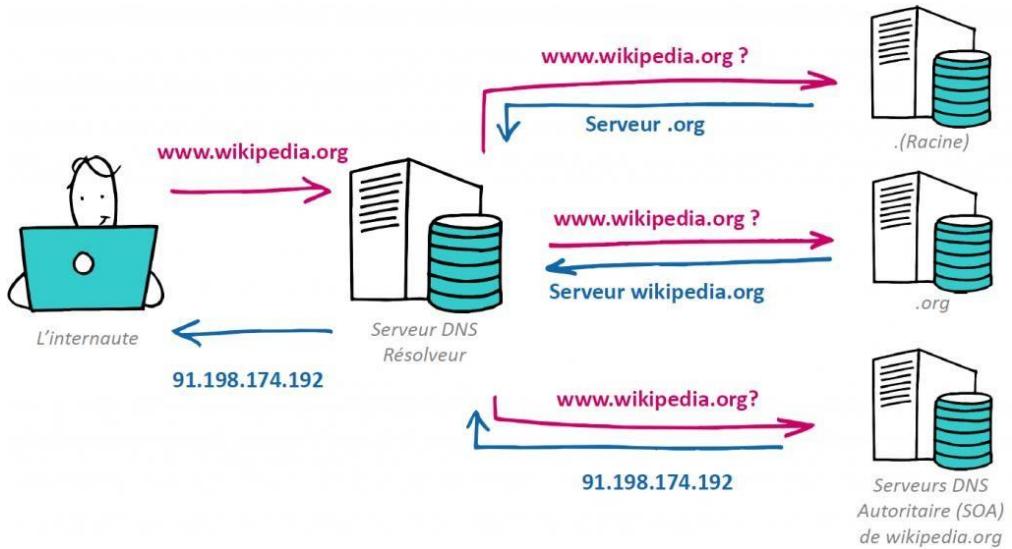
- Pour faciliter la recherche d'un site donné sur Internet, le système de noms de domaine (DNS) a été inventé. Le DNS permet d'associer un nom compréhensible, à une adresse IP. On associe donc une adresse logique, le nom de domaine, à une adresse physique l'adresse IP.
- Le nom de domaine et l'adresse IP sont uniques. Le DNS permet à votre message d'atteindre son destinataire et non quelqu'un d'autre possédant un nom de domaine similaire. Il vous permet également de taper "www.hsup.uae.ma" sans avoir à saisir une longue adresse IP et d'accéder au site web approprié.

Résolution *DNS*

Lorsqu'un internaute a saisi une adresse dans son navigateur, c'est donc un serveur DNS qui a traduit cette adresse humainement compréhensible, en une adresse IP, compréhensible par les ordinateurs et les réseaux. L'adresse www.hsup.uae.ma est ainsi traduite en 192.168.1.109.

On appelle cela la "**résolution DNS**".

Ce temps est d'autant plus faible que le serveur est performant : CPU (processeur), accès disque, et RAM (mémoire) doivent être correctement dimensionnés.
L'architecture logique du DNS est calquée sur la structure hiérarchique des noms de domaine.



Le DNS d'un niveau hiérarchique donné « délègue » au niveau inférieur le soin de traiter le sous-domaine suivant, jusqu'au dernier niveau qui, lui, connaît l'adresse IP correspondante au nom de domaine demandé.

Configuration DNS dans Notre Projet

Dans ce tutoriel, comment installer et configurer DNS (Bind9) sur le serveur Ubuntu 22.04. DNS est un service très important pour l'Internet local et public en raison de son utilisation pour le résolveur de noms comme la conversion IP en nom.

Etape1- Vérifier la version du serveur :

```
$ lsb_release -a && ip r
$ apt update && apt upgrade -y
```

```
abderrahane@dns:~$ lsb_release -a && ip r
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.2 LTS
Release:        22.04
Codename:       jammy
default via 192.168.1.1 dev wlp2s0 proto dhcp metric 600
169.254.0.0/16 dev wlp2s0 scope link metric 1000
192.168.1.0/24 dev wlp2s0 proto kernel scope link src 192.168.1.5 metric 600
abderrahane@dns:~$ sudo apt update && apt upgrade -y
```

Etape2- Installation Configure DNS (Bind9) service :

```
$ sudo apt install bind9  
$sudo apt-get install bind9utils bind9-doc
```

```
abderrahmane@dns:~$ sudo apt install bind9  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
bind9 is already the newest version (1:9.18.12-0ubuntu0.22.04.1).  
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.  
abderrahmane@dns:~$ sudo apt-get install bind9utils bind9-doc
```

```
abderrahmane@dns:~$ sudo nano /etc/resolv.conf
```

```
GNU nano 6.2                               /etc/resolv.conf  
# This is /run/systemd/resolve/resolv.conf managed by man:systemd-resolved(8).  
# Do not edit.  
#  
# This file might be symlinked as /etc/resolv.conf. If you're looking at  
# /etc/resolv.conf and seeing this text, you have followed the symlink.  
#  
# This is a dynamic resolv.conf file for connecting local clients directly to  
# all known uplink DNS servers. This file lists all configured search domains.  
#  
# Third party programs should typically not access this file directly, but only  
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a  
# different way, replace this symlink by a static file or a different symlink.  
#  
# See man:systemd-resolved.service(8) for details about the supported modes of  
# operation for /etc/resolv.conf.  
  
nameserver 192.168.1.5  
nameserver 8.8.8.8  
search hsup.uae.ma
```

Apres faire la commande suivant pour acceder au bind :

```
$ cd /etc/bind/
```

```
abderrahmane@dns:~$ cd /etc/bind/  
abderrahmane@dns:/etc/bind$ ls  
bind.keys  db.127  db.empty  named.conf          named.conf.local    rndc.conf  zones  
db.0        db.255  db.local   named.conf.default-zones  named.conf.options  rndc.key   zones.rfc1918  
abderrahmane@dns:/etc/bind$
```

On commence par les deux premiers

```
abderrahmane@dns:/etc/bind$ sudo nano named.conf.options
abderrahmane@dns:/etc/bind$ sudo nano named.conf.local
```

Le fichier named.conf.options

Apres fichier named.conf.local

```
GNU nano 6.2                                     named.conf.options
acl "Trusted" {
    192.168.1.5;      # ns1 - can be set to localhost
};

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.

    recursion yes;
    allow-recursion { trusted; };
    listen-on { 192.168.1.5; };
    allow-transfer { none; };

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    // 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;

    listen-on-v6 { any; };
};
```

Copier ce code mais avec vous donnees

```

GNU nano 6.2                                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 "hsup.uae.ma" {
    type master;
    file "/etc/bind/zones/db.fwd.hsup.uae.ma"; # zone file path
    allow-transfer { 192.168.1.5; };           # ns/www private IP address
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.rev.hsup.uae.ma"; # 192.168.0.0/16 subnet
    allow-transfer { 191.168.1.5; };           # ns/www private IP address
};


```

.1 Zone direct .2 Zone inverse

Zone direct (1) :

Le code que vous avez fourni est une configuration simplifiée pour une zone DNS dans le serveur BIND. Il indique que le serveur DNS est responsable de la zone "hsup.uae.ma" et que les enregistrements DNS de cette zone sont stockés dans le fichier "/etc/bind/hsup". Le serveur répondra aux requêtes DNS pour ce domaine en fournissant les enregistrements appropriés.

Zone inverse (2) :

Ce code configure une zone DNS inverse pour le sous-réseau 192.168.1.5/16. Le serveur DNS BIND est défini en tant qu'autorité principale pour cette zone, et les enregistrements sont stockés dans le fichier "/etc/bind/zones/db.rev.hsup.uae.ma". Seule l'adresse IP 192.168.1.5 est autorisée à effectuer des transferts de zone.

Maintenant on va créer un fichier repertoire qui contient les zones :

```
abderrahmane@dns:/etc/bind$ mkdir zones
```

Dans la Zone direct :

```
abderrahmane@dns:/etc/bind/zones$ sudo nano db.fwd.hsup.uae.ma
```

Insérez ci-dessous tout le texte dans ce fichier (mais avec votre données).

```
GNU nano 6.2                                         db.fwd.hsup.uae.ma
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.hsup.uae.ma. root.ns.hsup.uae.ma. (
                      2           ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                     2419200     ; Expire
                     604800 )     ; Negative Cache TTL
;
; name servers - NS records
      IN      NS      ns.hsup.uae.ma.

; name servers - A records
ns.hsup.uae.ma.          IN      A      192.168.1.5
www.hsup.uae.ma.         IN      A      192.168.1.5
;
```

Créez ensuite un fichier de zone inversée.

```
abderrahmane@dns:/etc/bind/zones$ sudo nano db.rev.hsup.uae.ma
```

Insérez ci-dessous tout le texte dans ce fichier.

```
GNU nano 6.2                                         db.rev.hsup.uae.ma
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.hsup.uae.ma. root.ns.hsup.uae.ma. (
                      5           ; Serial
                      604800      ; Refresh
                      86400       ; Retry
                     2419200     ; Expire
                     604800 )     ; Negative Cache TTL
;
; name servers
      IN      NS      ns.hsup.uae.ma.

5       IN      PTR      ns.hsup.uae.ma.
5       IN      PTR      www.hsup.uae.ma.
```

Puis retour d'un répertoire

```
$ cd ..
```

Ensuite, affichez rndc.key info à l'aide de la commande cat.

```
abderrahmane@dns:/etc/bind/zones$ cd ..
abderrahmane@dns:/etc/bind$ sudo cat rndc.key
key "rndc-key" {
    algorithm hmac-sha256;
    secret "7Wn+ysaFOKe5USAagsQK4aYUPOFydeYrObo0tlMfpSY=";
};
```

Créez maintenant le fichier rndc.conf.

```
$ sudo nano rndc.conf
```

Insérez ensuite ci-dessous tout le texte dans ce fichier.

```
GNU nano 6.2                                     rndc.conf
key "rndc-key" {
    algorithm hmac-sha256;
    secret "7Wn+ysaFOKe5USAagsQK4aYUPOFydeYrObo0tlMfpSY=";
};

options {
    default-key "rndc-key";
    default-server 127.0.0.1;
    default-port 953;
};
```

Etape3- Autorisation d'installation et redémarrage du service bind9.

Excuter les commandes suivants, il faut apres redemmarage etre Active.

```
abderrahmane@dns:/etc/bind$ sudo chown root:bind -R /etc/bind
abderrahmane@dns:/etc/bind$ sudo chown bind:bind -R /etc/bind/rndc.conf
abderrahmane@dns:/etc/bind$ sudo service bind9 restart
abderrahmane@dns:/etc/bind$ sudo service bind9 status
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-05-17 15:35:35 +01; 8s ago
     Docs: man:named(8)
     Process: 14837 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 14838 (named)
       Tasks: 10 (limit: 9277)
      Memory: 7.7M
        CPU: 147ms
       CGroup: /system.slice/named.service
               └─14838 /usr/sbin/named -u bind

May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:7fe::53#>
May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:7fd::1#53
May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:503:ba3e#>
May 17 15:35:35 dns.hsup.uae.ma named[14838]: network unreachable resolving './NS/IN': 2001:dc3::35#>
May 17 15:35:35 dns.hsup.uae.ma named[14838]: zone 255.in-addr.arpa/IN: loaded serial 1
May 17 15:35:35 dns.hsup.uae.ma named[14838]: all zones loaded
May 17 15:35:35 dns.hsup.uae.ma named[14838]: running
May 17 15:35:35 dns.hsup.uae.ma systemd[1]: Started BIND Domain Name Server.
May 17 15:35:35 dns.hsup.uae.ma named[14838]: managed-keys-zone: Key 20326 for zone . is now trusted
May 17 15:35:35 dns.hsup.uae.ma named[14838]: resolver priming query complete: success
lines 1-22/22 (END)
```

Etape4- Tester service bind9.

Maintenant, testez le service bind9 à partir du serveur.

```
abderrahmane@dns:/etc/bind$ cd  
abderrahmane@dns:~$ dig ns.hsup.uae.ma  
  
; <>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <>> ns.hsup.uae.ma  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1686  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: e61f343a8f160ad5010000006464e67554de810532720c57 (good)  
;; QUESTION SECTION:  
;ns.hsup.uae.ma.           IN      A  
  
;; ANSWER SECTION:  
ns.hsup.uae.ma.      604800  IN      A      192.168.1.5  
  
;; Query time: 0 msec  
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)  
;; WHEN: Wed May 17 15:36:37 +01 2023  
;; MSG SIZE  rcvd: 87
```

```
abderrahmane@dns:~$ dig www.hsup.uae.ma  
  
; <>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <>> www.hsup.uae.ma  
;; global options: +cmd  
;; Got answer:  
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36316  
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 1232  
; COOKIE: e2af827fc4faae3d010000006464e67a6f02cef76f7da193 (good)  
;; QUESTION SECTION:  
;www.hsup.uae.ma.          IN      A  
  
;; ANSWER SECTION:  
www.hsup.uae.ma.      604800  IN      A      192.168.1.5  
  
;; Query time: 0 msec  
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)  
;; WHEN: Wed May 17 15:36:42 +01 2023  
;; MSG SIZE  rcvd: 88
```

Vérifiez le résolveur PTR en utilisant la commande ci-dessous.

```
abderrahmane@dns:~$ dig -x 192.168.1.5

; <>> DiG 9.18.12-0ubuntu0.22.04.1-Ubuntu <>> -x 192.168.1.5
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31497
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 62146f08afaa4f0b010000006464e6c03d4fe12600661596 (good)
;; QUESTION SECTION:
;5.1.168.192.in-addr.arpa.      IN      PTR

; ANSWER SECTION:
.1.168.192.in-addr.arpa. 604800 IN      PTR      www.hsup.uae.ma.
.1.168.192.in-addr.arpa. 604800 IN      PTR      ns.hsup.uae.ma.

;; Query time: 0 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Wed May 17 15:37:52 +01 2023
;; MSG SIZE rcvd: 127
```

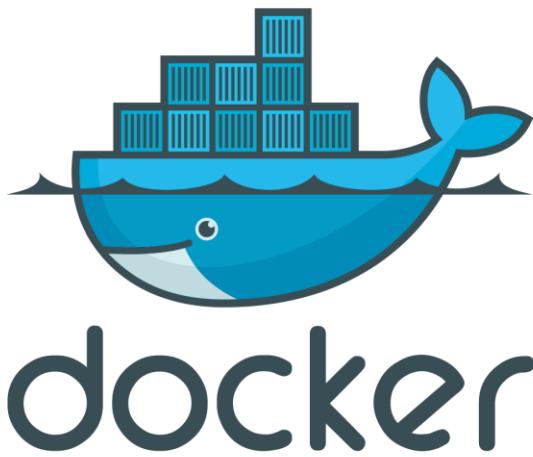
Vérifier le test de nom.

```
abderrahmane@dns: $ nslookup www
Server:      192.168.1.5
Address:    192.168.1.5#53

Name:  www.hsup.uae.ma
Address: 192.168.1.5
```

Vérifier le test ip

```
abderrahmane@dns:~$ nslookup 192.168.1.5
5.1.168.192.in-addr.arpa      name = ns.hsup.uae.ma.
5.1.168.192.in-addr.arpa      name = www.hsup.uae.ma.
```



1. Introduction

- **Docker** est une application qui simplifie le processus de gestion des processus d'application dans des conteneurs. Les conteneurs vous permettent d'exécuter vos applications dans des processus isolés des ressources. Ils sont similaires aux machines virtuelles, mais les conteneurs sont plus portables, plus respectueux des ressources et plus dépendants du système d'exploitation hôte.
- **Étape 1: Installation de "Docker"**



Bash Terminal (Ubuntu)

```
> sudo apt update // mettre à jour les packets  
> sudo apt install apt-transport-https ca-certificates  
curl software-properties-common
```

Ensuite, installez quelques packages prérequis qui permettent à **apt** d'utiliser des packages via **HTTPS** :

- Ajoutez ensuite la clé GPG du référentiel Docker officiel à notre système :



Bash Terminal (Ubuntu)

```
> curl -fsSL  
https://download.docker.com/linux/ubuntu/gpg | sudo  
gpg --dearmor -o /usr/share/keyrings/docker-archive-  
keyring.gpg
```

- Ajoutez le dépôt Docker aux sources APT :



Bash Terminal (Ubuntu)

```
> echo "deb [arch=$(dpkg --print-architecture) signed-  
by=/usr/share/keyrings/docker-archive-keyring.gpg]  
https://download.docker.com/linux/ubuntu $(lsb_release -  
cs) stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null
```

- Nous mettons à jour à nouveau notre liste de packages existante pour que l'ajout soit reconnu :

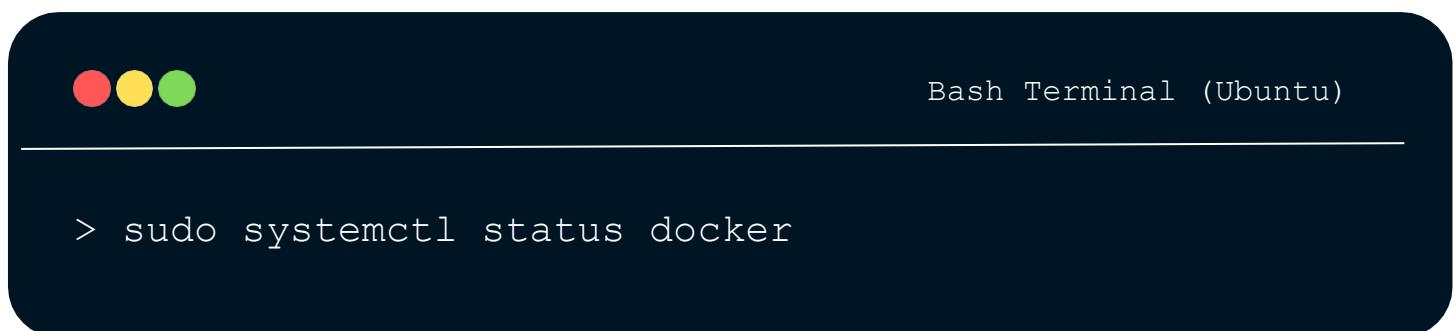


Bash Terminal (Ubuntu)

```
> sudo apt update // mettre à jour les packets  
> apt-cache policy docker-ce // installer à partir du  
Repo Docker  
  
> sudo apt install docker-ce // Enfin, installez  
Docker
```

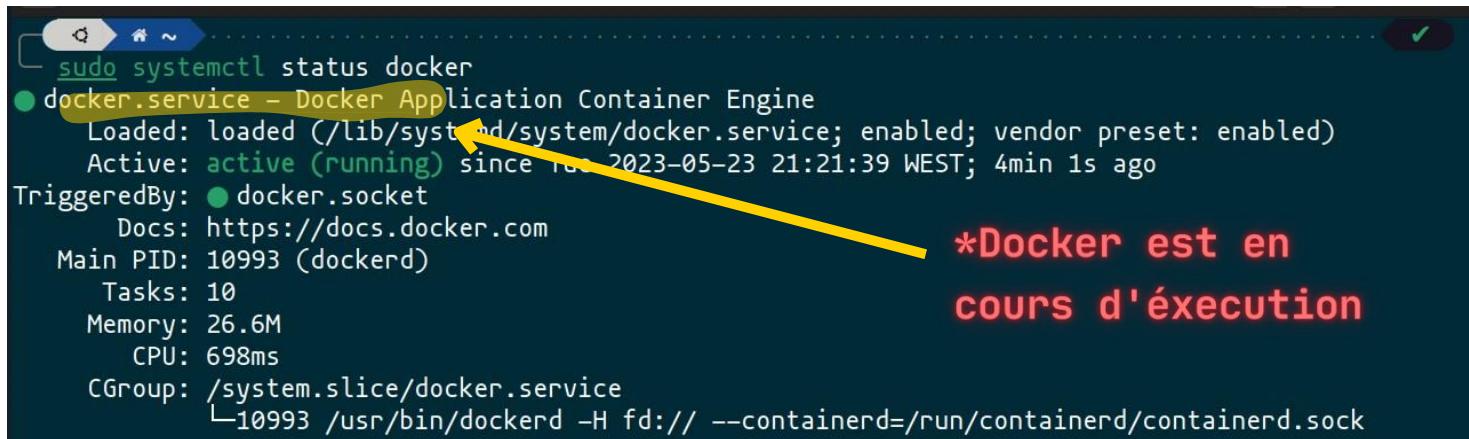
```
sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  apache2-bin apache2-data apache2-utils libaprpri1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
  docker-compose-plugin libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-rootless-extras
  docker-compose-plugin libslirp0 pigz slirp4netns
0 upgraded, 9 newly installed, 0 to remove and 158 not upgraded.
Need to get 110 MB of archives.
```

- Docker devrait maintenant être installé, le **daemon** démarré et le processus activé pour démarrer au démarrage. On vérifie qu'il est en **cours d'exécution** :



Bash Terminal (Ubuntu)

```
> sudo systemctl status docker
```



```
sudo systemctl status docker
● docker.service - Docker Application Container Engine
  Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2023-05-23 21:21:39 WEST; 4min 1s ago
    TriggeredBy: ● docker.socket
      Docs: https://docs.docker.com
     Main PID: 10993 (dockerd)
       Tasks: 10
      Memory: 26.6M
        CPU: 698ms
       CGroup: /system.slice/docker.service
                 └─10993 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

*Docker est en cours d'exécution

- L'installation de Docker nous donne désormais non seulement le service **Docker (daemon)**, mais également l'utilitaire de **ligne de commande Docker** ou **le client Docker**.

• Étape 2: Utilisation de la commande Docker

- Utiliser docker consiste à lui passer une chaîne d'options et de commandes suivies d'arguments. La syntaxe prend cette forme :



Bash Terminal (Ubuntu)

```
> docker [option] [command] [arguments]
```

```
└── docker
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Common Commands:
  run      Create and run a new container from an image
  exec    Execute a command in a running container
  ps       List containers
  build   Build an image from a Dockerfile
  pull    Download an image from a registry
  push    Upload an image to a registry
  images  List images
  login   Log in to a registry
  logout  Log out from a registry
  search  Search Docker Hub for images
  version Show the Docker version information
  info    Display system-wide information
```

• Étape 3: Travailler avec des images Docker

- Les conteneurs Docker sont construits à partir d'**images Docker**. Par défaut, **Docker** extrait ces images de **Docker Hub**, un registre Docker géré par Docker, la société à l'origine du projet Docker. N'importe qui peut héberger ses images Docker sur Docker Hub, donc la plupart des applications et des distributions Linux dont vous aurez besoin auront des images hébergées là-bas.
- Pour vérifier si nous pouvons accéder et télécharger des images depuis **Docker Hub**, nous utilisons :



```
> docker run hello-
```

- La sortie indiquera que Docker fonctionne correctement :

```
docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:fc6cf906cbfa013e80938cdf0bb199fbdbb86d6e3e013783e5a766f50f5dbce0
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

- L'image **hello_world** :

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
606748c93abd	hello-world	"/hello"	52 minutes ago	Exited (0) 52 minutes ago		funny_wilson

• Exécution d'un conteneur Docker



```
> docker pull ubuntu:latest
```

```
> docker run -it ubuntu
```

- Par exemple, exécutons un conteneur en utilisant la dernière image d'**Ubuntu**. La combinaison des commutateurs **-i** et **-t** vous donne un accès shell interactif au conteneur :

```
[~] docker pull ubuntu:latest
latest: Pulling from library/ubuntu
dbf6a9befcde: Pull complete
Digest: sha256:dfd64a3b4296d8c9b62aa3309984f8620b98d87e47492599ee20739e8eb54fbf
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

```
[~] docker run -it ubuntu
root@77bb36931a86:/# ls
bin dev home lib32 libx32 mnt proc run srv tmp var
boot etc lib lib64 media opt root sbin sys usr
root@77bb36931a86:/#
```

• Gestion des conteneurs Docker

- Pour afficher tous les conteneurs - actifs et inactifs, exédez **docker ps** avec **-a** :



Bash Terminal (Ubuntu)

```
> docker ps -a
```

- Pour afficher le dernier conteneur que vous avez créé, passez-lui le **-l**



Bash Terminal (Ubuntu)

```
> docker ps -l
```

```
[~] docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
2e727b053eb2 ubuntu "/bin/bash" 17 seconds ago Up 16 seconds
77bb36931a86 ubuntu "/bin/bash" 7 minutes ago Exited (0) 2 minutes ago
[silky_pare
[elastic_sand]
```

- Pour démarrer un conteneur arrêté, utilisez docker start, suivi de l'ID du conteneur ou du nom du conteneur. Commençons le conteneur basé sur Ubuntu avec l'ID **2e727b053eb2** :



Bash Terminal (Ubuntu)

```
> docker start 2e727b053eb2
```

- Pour arrêter un conteneur en cours d'exécution, utilisez docker stop, suivi de l'ID ou du nom du conteneur. Cette fois, nous utiliserons le nom que Docker a attribué au conteneur, qui est **silly_pare**



Bash Terminal (Ubuntu)

```
> docker stop silly_pare
```

A screenshot of a terminal window on a dark background. The window title is 'Bash Terminal (Ubuntu)'. In the terminal, the command 'docker stop silly_pare' is being typed. The output shows 'silly_pare' followed by a new line. The terminal has a standard Linux-style interface with tabs at the bottom.

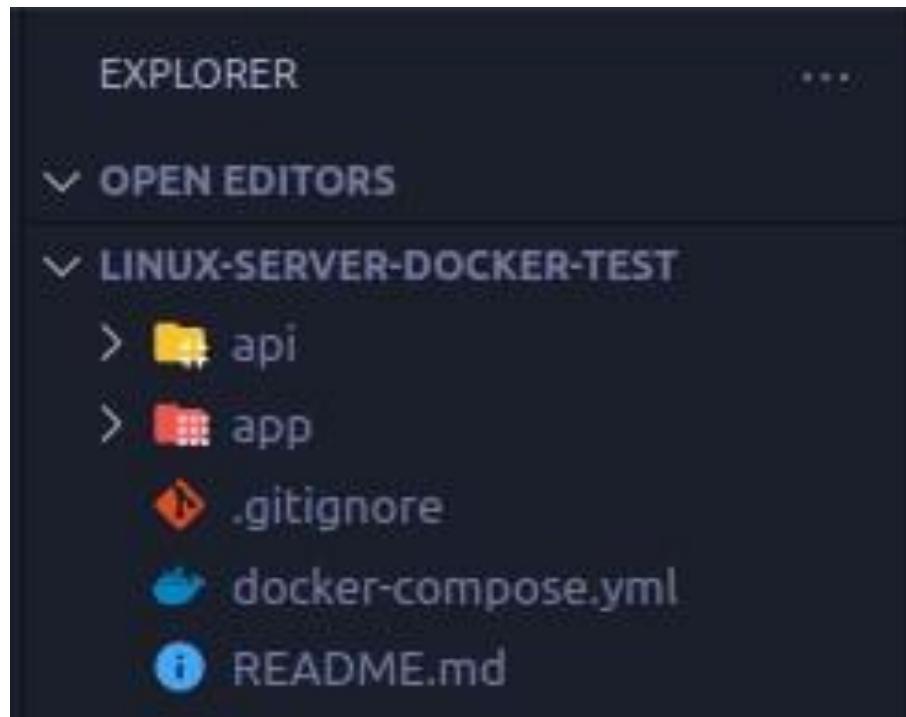
- Une fois que nous avons décidé que nous n'avions plus besoin d'un conteneur, nous pouvons le supprimer avec la commande docker rm, en utilisant à nouveau soit l'ID du conteneur, soit le nom.



Bash Terminal (Ubuntu)

```
> docker rm silly_pare
```

- **Étape 4: Dockerizing notre projet**



1. Configuration du conteneur postgres

-> Pourquoi utiliser Postgres avec docker pour le développement local?

Il existe de nombreuses bonnes raisons d'utiliser n'importe quelle base de données comme Postgres avec Docker pour le développement local, voici quelques bonnes raisons :

1. L'utilisation de plusieurs versions de PostgreSQL selon le projet ou tout autre besoin est très simple.
2. Généralement, avec docker, s'il s'exécute sur votre machine, il fonctionnera pour votre ami, sur un environnement intermédiaire et un environnement de production, étant donné que la compatibilité des versions est maintenue.
3. Lorsqu'un nouveau membre de l'équipe se joint, le nouveau membre peut démarrer en quelques heures, il ne faut pas des jours pour être productif.

- **PostgreSQL with docker-compose**

```
docker-compose.yml x
  docker-compose.yml
  31
  32 db:
  33   container_name: db
  34   image: postgres:13
  35   restart: always
  36   hostname: db
  37   ports:
  38     - "5432:5432"
  39
  40   environment:
  41     - POSTGRES_USER=postgres
  42     - POSTGRES_PASSWORD=postgres
  43     - POSTGRES_DB=postgres
  44     - POSTGRES_HOST=db
  45
  46   volumes:
  47     - pgdata:/var/lib/postgresql/data
  48
  49
  50 volumes:
  51   pgdata: {}
```

- **container_name: db** - le nom du conteneur.
- **image: postgres:13** - Nous utilisons la version 13 de postgres.
- **restart: always** - toujours redémarrer le conteneur en cas d'erreur.
- **hostname: db** - hostname associé à la base de données
- **ports: 5432:5432** - le port sur lequel la base de données écoutera les événements
- **environment:**
 - **POSTGRES_USER=postgres**
 - **POSTGRES_PASSWORD=postgres**
 - **POSTGRES_DB=postgres**
 - **POSTGRES_HOST=db**

Les paramètres requis pour connecter la base de données postgres, y compris le nom d'utilisateur, le mot de passe et le nom de la base de données.

- Pour démarrer la base de données **PostgreSQL** à l'aide de ce fichier **docker-compose.yml**, vous pouvez exécuter la commande suivante :

Bash Terminal (Ubuntu)

```
> docker-compose up -d
```

```
~/Downloads/Linux-Server-docker-test/test
docker-compose up -d
Creating network "test_default" with the default driver
Creating volume "test_pgdata" with default driver
Creating db ... done
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
4a0d413d0f34	postgres:13	"docker-entrypoint.s..."	About a minute ago	Up About a minute	0.0.0.0:5432->5432/tcp
p	:::5432->5432/tcp	db			

- on se connecte à la base de données via la commande : **docker exec -it db psql -U postgres**
- nous créons une table à des fins de test appelée "accounts"

```
~/Downloads/Linux-Server-docker-test/test
docker exec -it db psql -U postgres
psql (13.11 (Debian 13.11-1.pgdg110+1))
Type "help" for help.

postgres=# CREATE TABLE accounts (
    user_id serial PRIMARY KEY,
    username VARCHAR ( 50 ) UNIQUE NOT NULL,
    password VARCHAR ( 50 ) NOT NULL,
    email VARCHAR ( 255 ) UNIQUE NOT NULL,
    created_on TIMESTAMP NOT NULL,
    last_login TIMESTAMP
);
CREATE TABLE
postgres=#

```

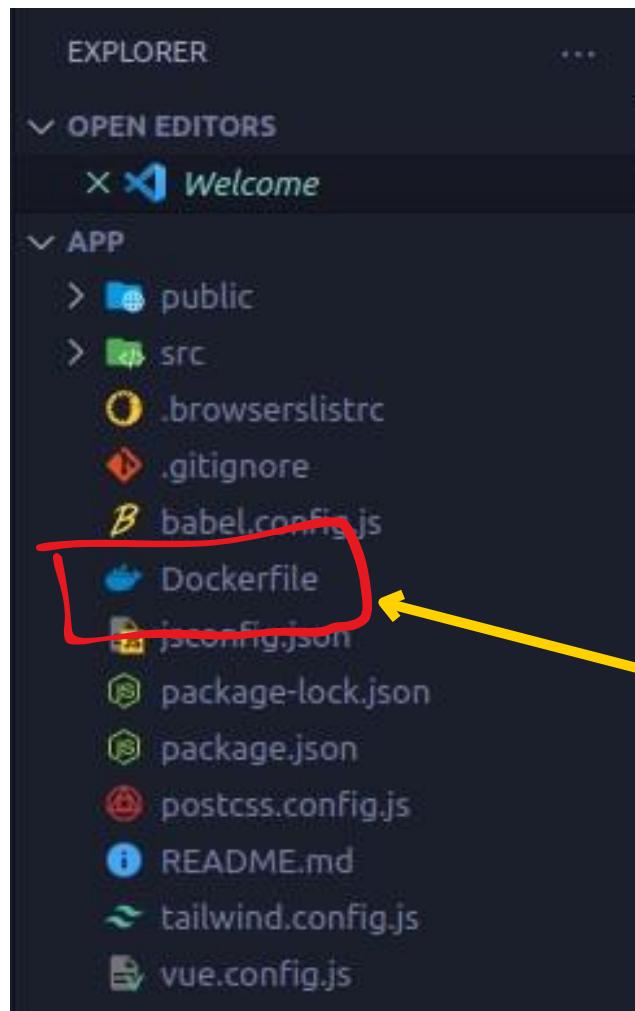
```
postgres=# \dt
      List of relations
 Schema |   Name    | Type  | Owner
-----+-----+-----+-----
 public | accounts | table | postgres
(1 row)

postgres=#

```

2.Configuration du conteneur vueJS

- Commençons par créer un fichier **Dockerfile** dans le dossier racine de notre projet



Le fichier
Dockerfile de
Configuration
de vueJS

- Le fichier Dockerfile:

```
FROM node:lts-alpine
RUN npm install -g http-server
WORKDIR /app
COPY package*.json .
RUN npm install
COPY . .
RUN npm run build
CMD [ "npm", "start" ]
```

The screenshot shows a code editor window displaying a Dockerfile. The file starts with a base image 'node:lts-alpine', installs 'http-server' using 'npm', sets the working directory to '/app', copies the package files, installs dependencies, copies the current directory, runs a build script, and finally starts the application with 'npm start'.

- **FROM node:its-alpine**
 - Choisissez l'image sur laquelle Node est déjà installé
- **RUN npm install -g http-server**
 - installer un serveur http simple pour servir du contenu statique
- **WORKDIR /app**
 - créer du dossier 'app' le répertoire de travail actuel
- **COPY package*.json ./**
 - copier à la fois 'package.json' et 'package-lock.json' (si disponible)

- **RUN npm install**
 - installer les dépendances du projet
- **COPY ..**
 - copier les fichiers et dossiers du projet dans le répertoire de travail actuel (c'est-à-dire le dossier 'app')
- **RUN npm run build**
 - créer une application pour la production avec minification
- **CMD ["npm", "start"]**
 - starting npm

```

  docker-compose.yml ×
  docker-compose.yml
  1   version: "3.9"
  2
  3   services:
  4     web:
  5       image: test_website
  6       container_name: test_website
  7       build:
  8         context: app
  9         dockerfile: ./Dockerfile
 10       ports:
 11         - 8080:8080
 12       volumes:
 13         - ./app:/app
 14       command: npm run serve

```

- **container_name:** test_website - le nom du conteneur.
- **image:** test_website - Nous créons une image de l'app.
- **build:**
 - context: app
 - dockerfile: ./Dockerfile
- L'emplacement du fichier Dockerfile de configuration et le contexte de l'app

- **ports: 8080:8080** - le port sur lequel l'app vueJS écoutera les événements
- **volumes:**

- ./app:/app

- Il monte un volume sur ./app sur la machine hôte.

- **command:**
 - npm run serve
- la commande pour lancer l'app vuejs sur l'adresse "0.0.0.0:8080"

```
~ ~/Downloads/Linux-Server-docker-test
docker-compose up --build web
Creating network "linux-server-docker-test_default" with the default driver
Creating volume "linux-server-docker-test_pgdata" with default driver
Building web
[+] Building 8.6s (4/11)
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 210B
  => [internal] load metadata for docker.io/library/node:lts-alpine
  => [1/7] FROM docker.io/library/node:lts-alpine@sha256:1ccc70acda680aa4ba47f53e7c40b2d4d6892de74817128e0662d
  => => resolve docker.io/library/node:lts-alpine@sha256:1ccc70acda680aa4ba47f53e7c40b2d4d6892de74817128e0662d
  => => sha256:1ccc70acda680aa4ba47f53e7c40b2d4d6892de74817128e0662d
  => => sha256:44aaf1cc80eaed6572a0f2ef7d6b5a2982d54481e4255480041ac92221e2f11
  => => sha256:6f44d13dd2586511d630d7708da1eaac09c05693d9e3e06f3f72206927d3201d
  => => sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c4e5add29d7f64b87abdaa09
  => => sha256:931b0e865bc24d4d3f7032c4a9b98b4790ee7bb8dc0d2fd02e15e163d806034
  => => sha256:60542df8b663b3c876b9fd0ad8459ac628a50bd8684149e54c3df2e44397f03
  => => extracting sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c4e5add29d7f64b87abdaa09
```

- Pour démarrer l'application **VueJS** à l'aide de ce fichier **docker-compose.yml**, vous pouvez exécuter la commande suivante :

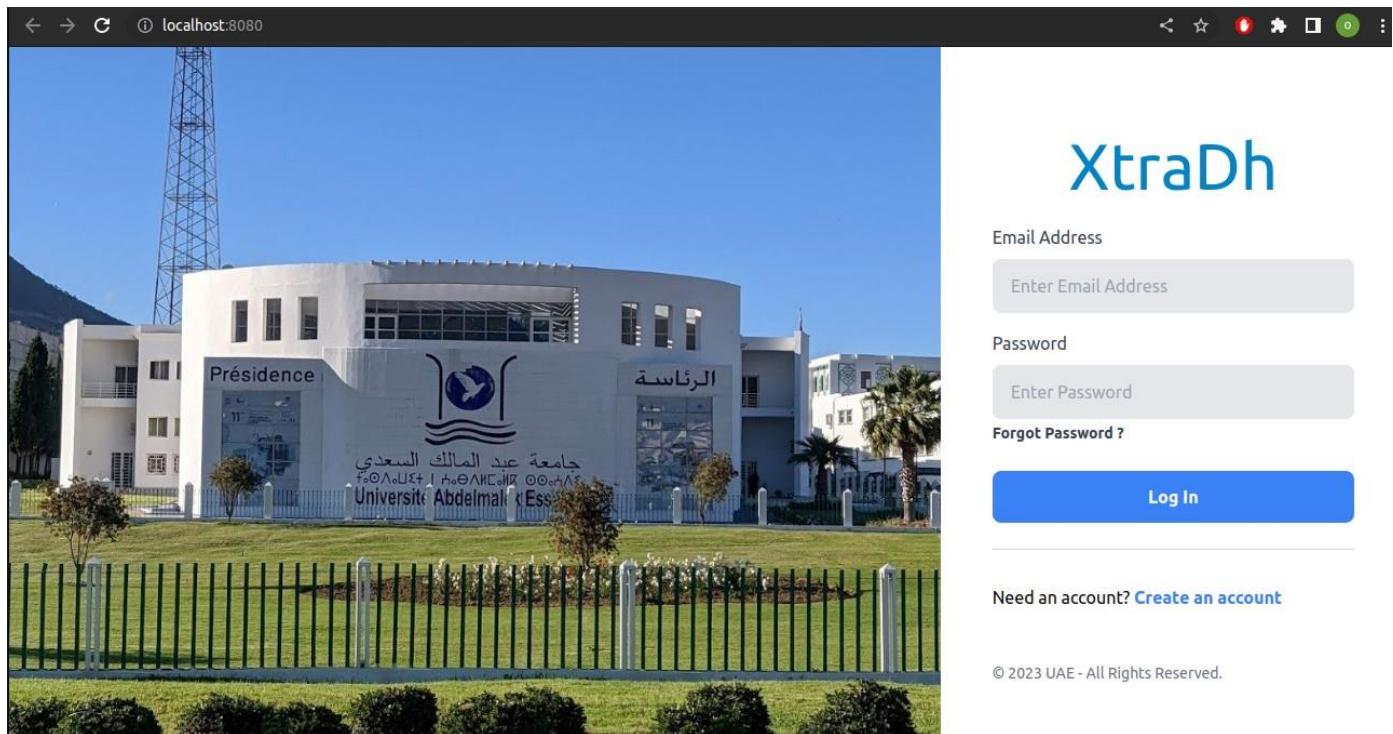


Bash Terminal

> docker-compose up --build

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
ee1678910c35	test_website	"docker-entrypoint.s..."	9 minutes ago	Exited (127) 4 minutes ago		test_w

- On vérifie sur le navigateur: <http://localhost:8080/>



3. Configuration du conteneur Laravel API

- Commençons par créer un fichier **Dockerfile** dans le dossier racine de notre projet

Le fichier ...

```
EXPLORER Dockerfile
OPEN EDITORS Dockerfile
API
  app
  bootstrap
  config
  database
  public
  resources
  routes
  storage
  tests
  .editorconfig
  .env.example
  .gitattributes
```

Dockerfile

```
.gitignore
artisan
composer.json
composer.lock
docker-compose.debug.yml
Dockerfile
package.json
phpunit.xml
README.md
vite.config.js
```

~ 60 ~

- **Le fichier Dockerfile:**

```
Dockerfile
FROM php:8.1
RUN apt-get update && apt-get install -y \
    libpq-dev \
    && docker-php-ext-install pdo pdo_pgsql
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
WORKDIR /var/www/html
COPY . /var/www/html
ENV COMPOSER_ALLOW_SUPERUSER=1
RUN composer install
RUN chown -R www-data:www-data /var/www/html/storage /var/www/html/bootstrap/cache
CMD php artisan migrate && php artisan serve --host=0.0.0.0 --port=8000
```

- **FROM php:8.1**

- Nous utiliserons PHP version 8.1 comme image de base

- **RUN apt-get update && apt-get install -y **

- libpq-dev **

- && docker-php-ext-install pdo pdo_pgsql**

- installer les dépendances nécessaires pour se connecter à la base de

- données comme le pilote db etc...**

- **RUN curl -sS https://getcomposer.org/installer | php ---- install-**

- dir=/usr/local/bin --filename=composer**

- installer composer dans le conteneur pour exécuter l'application laravel

- **WORKDIR /var/www/html**

- définir le répertoire /var/www/html comme une répertoire racine de l'application laravel.

- **COPY ./var/www/html :** copier les fichiers et dossiers du projet dans le répertoire du travaille.

- ENV COMPOSER_ALLOW_SUPERUSER=1

- accorder des autorisations au composer pour exécuter l'application

laravel

- RUN composer install

- Finir l'installation de composer

- RUN chown -R www-data:www-data /var/www/html/storage
/var/www/html/bootstrap/cache

- accorder les permissions d'exécuter les sous-dossiers du répertoire de l'application

- CMD php artisan migrate && php artisan serve --host=0.0.0.0 --port=8000

- migration des fichiers de configuration de la base de données avant de démarrer l'application sur l'hôte local : 8000

- Le fichier docker-compose.yml:

```
api:  
  container_name: laravel_test_api  
  image: laravel_test_api  
  build:  
    context: api  
    dockerfile: ./Dockerfile  
  
  ports:  
    - "8000:8000"  
  env_file:  
    - api/.env  
  volumes:  
    - ./api:/var/www/html  
  depends_on:  
    - db
```

- container_name: laravel_test_api - le nom du conteneur.

- image: laravel_test_api - Nous créons une image de l'app.

- **build:**

- context: api
- dockerfile: ./Dockerfile

- L'emplacement du fichier Dockerfile de configuration et le contexte de l'app

- **ports: 8000:8000** - le port sur lequel l'app Laravel écoutera les événements

- **volumes:**

- ./api:/var/www/html

- Il monte un volume sur ./api sur la machine hôte.

- **env_file:**

- api/.env

- L'emplacement du fichier .env qui contient les paramètres de connexion à la base de données.

- **depends:**

- db

- L'API du conteneur dépend de la fonction du conteneur de la base de données et ne démarrera que si le conteneur bd est opérationnel et fonctionne correctement.

- Pour démarrer l'application Laravel à l'aide de ce fichier **docker-compose.yml**, vous pouvez exécuter la commande suivante :



Bash Terminal (Ubuntu)

```
> docker-compose up --build api
```

```
docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
NAMES
2e6cc8dcdf02      laravel_test_api   "docker-php-entrypoi..."   55 seconds ago    Up 54 seconds    0.0.0.0:8000->8000/tcp,
:::8000->8000/tcp
28b22e1df634      postgres:13       "docker-entrypoint.s..."  15 minutes ago   Up 5 minutes    0.0.0.0:5432->5432/tcp,
:::5432->5432/tcp
db
```

```

[~/Downloads/Linux-Server-docker-test] docker-compose up --build api
Building api
[+] Building 16.9s (4/11)
=> [internal] load build definition from Dockerfile          0.1s
=> => transferring dockerfile: 518B                         0.0s
=> [internal] load .dockerignore                            0.1s
=> => transferring context: 2B                           0.0s
=> [internal] load metadata for docker.io/library/php:8.1   2.6s
=> [1/7] FROM docker.io/library/php:8.1@sha256:1e2526b7fa73a1da7ebffa6773aee9a79c6e056031 14.1s
=> => resolve docker.io/library/php:8.1@sha256:1e2526b7fa73a1da7ebffa6773aee9a79c6e0560311 0.1s
=> => sha256:96844aaaa11691e3782ca2dceabd00e56fcfc7222c737b12806f93a0b50c26 2.20kB / 2.20kB 0.0s
=> => sha256:593ded604f117b0b52730cd26777b233477cae2d63a0f709e920d2818afdc 9.76kB / 9.76kB 0.0s
=> => sha256:662d8f2fcdb9ff5b9497538bdbb929dficc5f1d371a15f5ea427ade8ed4dc08f 224B / 224B 0.3s
=> => sha256:1e2526b7fa73a1da7ebffa6773aee9a79c6e05603110b496bb8737651e332 1.86kB / 1.86kB 0.0s
=> => sha256:78fe0ef5ed7711aab5b14ba79dee1d8f750f1cb661645280d76336db601 8.39MB / 91.64MB 13.9s

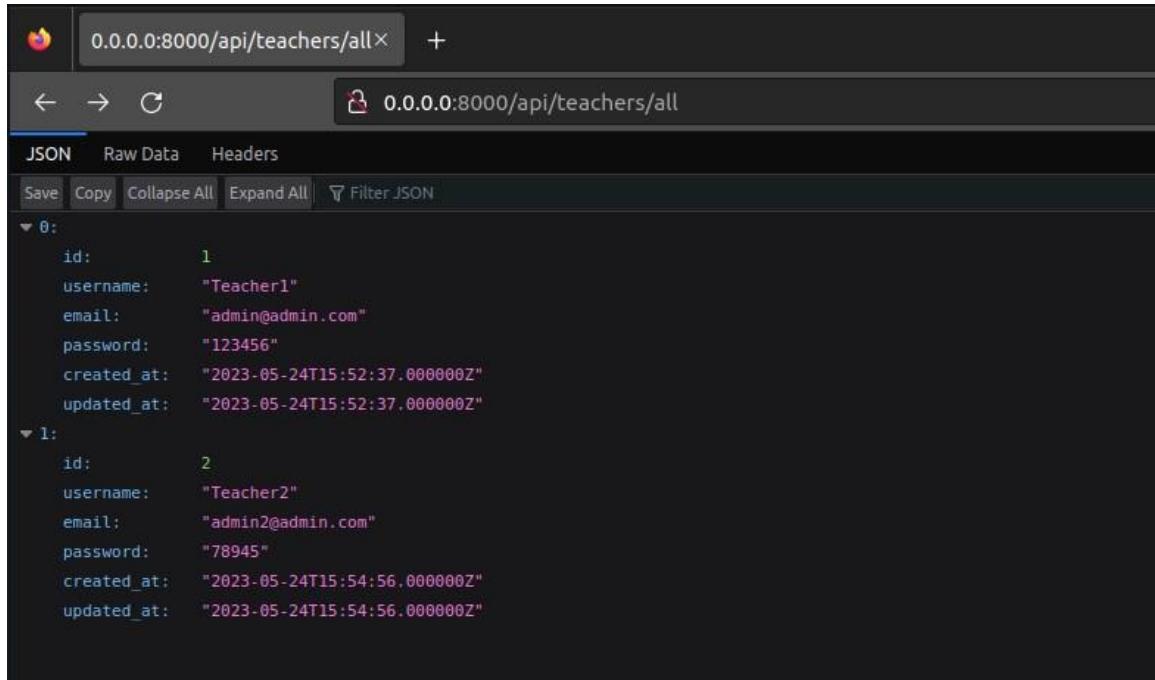
```

```

=> => naming to docker.io/library/laravel_test_api           0.0s
db is up-to-date
Recreating laravel_test_api ... done
Attaching to laravel_test_api
laravel_test_api | INFO  Preparing database.
laravel_test_api | Creating migration table ..... 15ms DONE
laravel_test_api | INFO  Running migrations.
laravel_test_api | 2014_10_12_000000_create_users_table ..... 28ms DONE
laravel_test_api | 2014_10_12_100000_create_password_reset_tokens_table ..... 27ms DONE
laravel_test_api | 2019_08_19_000000_create_failed_jobs_table ..... 43ms DONE
laravel_test_api | 2019_12_14_000001_create_personal_access_tokens_table ..... 45ms DONE
laravel_test_api | 2023_05_13_135305_create_teachers_table ..... 27ms DONE
laravel_test_api |
laravel_test_api | INFO  Server running on [http://0.0.0.0:8000].
laravel_test_api | Press Ctrl+C to stop the server
laravel_test_api |

```

• On vérifie sur le navigateur: <http://0.0.0.0:8000>/



4.Configuration du conteneur NGINX



```
# Use Nginx as the server
FROM nginx:latest

# Remove the default Nginx configuration
RUN rm /etc/nginx/conf.d/default.conf

# Copy your custom Nginx configuration file
COPY nginx/myconfig.conf /etc/nginx/conf.d/

# Copy the built files from the previous stage
COPY --from=build-stage /app/dist /usr/share/nginx/html

# Expose port 80
EXPOSE 80

# Start Nginx
CMD ["nginx", "-g", "daemon off;"]
```

- **FROM nginx:latest**
 - extraire l'image de la dernière version de nginx à partir du hub docker
- **RUN rm /etc/nginx/conf.d/default.conf**
 - Supprimer le fichier de configuration par défaut
- **COPY nginx/myconfig.conf /etc/nginx/conf.d/**
 - Copiez notre fichier de configuration Nginx personnalisé
- **COPY --from=build-stage /app/dist /usr/share/nginx/html**
 - Copiez les fichiers de construction à partir de l'application vue js
- **EXPOSE 80**
 - Utiliser le port 80
- **CMD ["nginx", "-g", "daemon off;"]**
 - lancer NGINX

- Le fichier docker-compose.yml:

```
14    nginx:
15      image: nginx:latest
16      ports:
17        - 80:80
18        - 443:443
19      volumes:
20        - ./nginx:/etc/nginx/conf.d
21        - ./ssl:/var/www/ssl
22      depends_on:
23        - app
```

- **image: nginx:latest** - la dernière image de nginx
- **ports:**
 - **80:80**
 - **443:443**
 - Utiliser 80 pour HTTP et 443 pour HTTPS
- **volumes:**
 - **./nginx:/etc/nginx/conf.d**
 - **./ssl:/var/www/ssl**
 - Copier le fichier de configuration et la certificat ssl
- **depends_on:**
 - **app**
 - Le conteneaire nginx dépend de l'application vuejs pour commencer.
- Pour démarrer l'applicuation **VueJS et NGINX** à l'aide de ce fichier **docker-compose.yml**, vous pouvez exécuter la commande suivante :

```
docker-compose up -d
Starting docker_app_1 ... done
Starting docker_nginx_1 ... done

docker-compose up
Starting docker_app_1 ... done
docker_nginx_1 is up-to-date
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
0377e945752c	nginx:latest	"/docker-entrypoint...."	55 seconds ago	Up 11 seconds	0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp docker_nginx_1
NAMES					

- On vérifie sur le navigateur: <https://hsup.uae.ma>

The image shows a dual-pane view. On the left is a photograph of a modern white building with a blue and white logo, identified as the University of Abdellah ibn Haldun. On the right is a screenshot of a web browser displaying a login form for 'XtraDh'. The form includes fields for 'Email Address' and 'Password', a 'Forgot Password?' link, and a 'Log In' button. Below the form, there is a link to 'Create an account'.



Configuration Routeur

- Nous allons configurer le routeur (un routeur TP Link dans ce cas) pour modifier les paramètres DNS afin de rendre notre nom de domaine "hsup.uae.ma" accessible à tous sur le réseau local
- **Etape1-Connecter à la page de configuration du routeur :**

L'address du
passerelle par
défaut
192.168.1.1

tp-link

300Mbps Wireless N ADSL2+ Modem Router
Model No. TD-W8960N

Username

Password

Log in

- On entre les données de connexion pour accéder au dashboard principal du routeur

Etape2-La configuration du routeur :



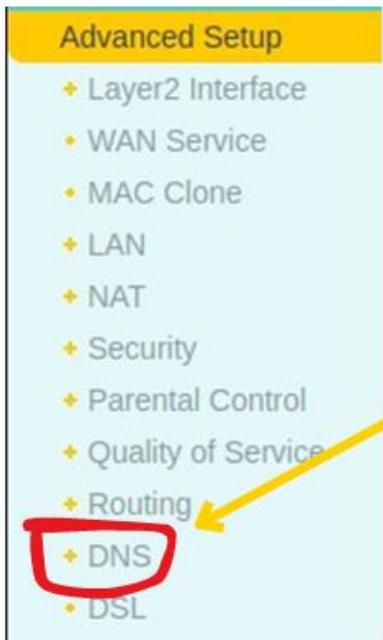
<input type="radio"/> Disable DHCP Server	<input checked="" type="radio"/> Enable DHCP Server				
Start IP Address:	192.168.1.100				
End IP Address:	192.168.1.200				
Leased Time (hour):	24 (1~48)				
Static IP Lease List: (A maximum 32 entries can be configured)					
MAC Address	IP Address	Status	Enable/Disable	Edit	Remove
3c:a0:67:1f:fd:b8	192.168.1.109	Enabled	<input type="button" value="Disable"/>	<input type="button" value="Edit"/>	<input type="checkbox"/>
<input type="button" value="Add"/> <input type="button" value="Enable All"/> <input type="button" value="Select All"/> <input type="button" value="Remove"/>					
<input type="button" value="Save/Apply"/>					

*On ajoute l'adresse MAC du serveur (PC)

*On l'assigne une adresse IP statique 192.168.1.109

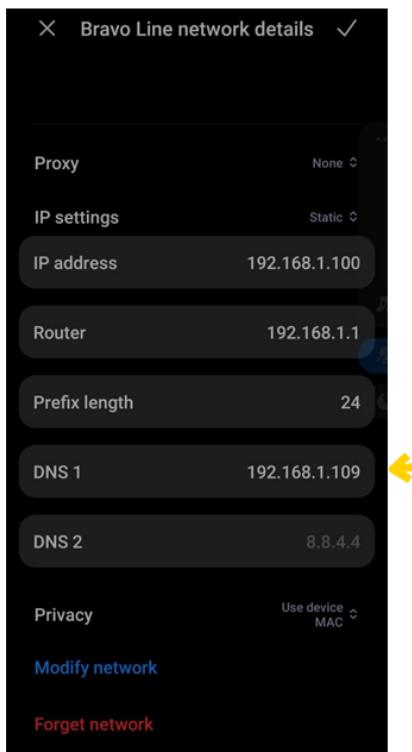
dans ce cas *Sauvegarder les changements

maintenant, nous configurons les paramètres DNS dans le routeur pour qu'il fonctionne comme un serveur DNS pour le site Web

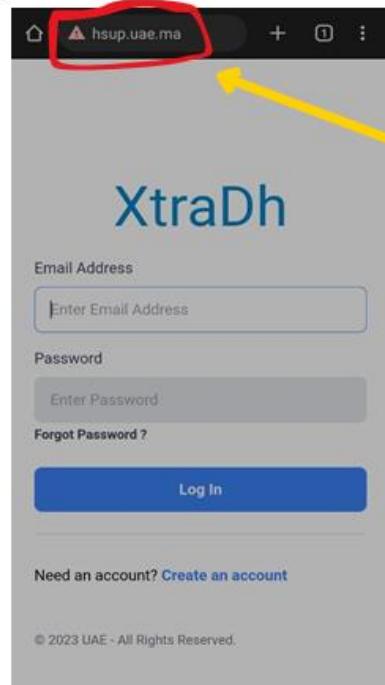


*modifier les paramètres de DNS

- Après ces changements, chaque appareil qui se connecte à ce réseau aura **192.168.1.109** comme DNS principal et pourra accéder au www.hsup.uae.ma

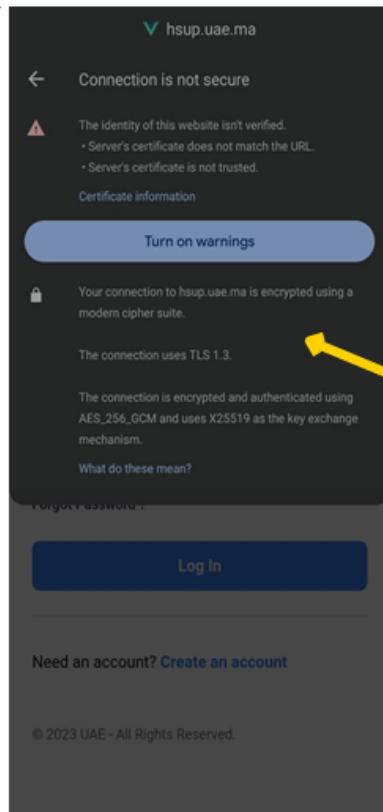


*DNS est bien configuré au niveau de réseau



*Le domaine
"hsup.uae.ma"
fonctionne
correctement

fonctionnalités https et ssl/tls fonctionnent également sans problème



*Les détails de la
certificat ssl/tls

