

PROJET D'INTÉGRATION DES ACQUIS DE LA PREMIÈRE ANNÉE GÉNIE INFORMATIQUE

RAPPORT D'APPRENTISSAGE & TECHNIQUE

Administration Système - Conteneurisation avec Docker

Présenté par :

SOUIDAK FIRDAOUS - AHMYTTOU AYA

Présenté à :

Membres du jury :

Pr. EL HADDAD Mohamed	ENSA de Tanger	-----
Pr. BELMOKADEM Houda	ENSA de Tanger	-----
Pr. Mohamed GHAILANI	ENSA de Tanger	Encadrant
M. BOUKTAIB Soufian	ENSA de Tanger	Etudiant Tuteur
M. EL AALOUCH MohamedYassine	ENSA de Tanger	Etudiant Tuteur

SOMMAIRE

DNS

1. Introduction	p 1-3
A- Qu'est-ce qu'un DNS ?	
B- Rôle du DNS	
C- La hiérarchie DNS	
2-Configuration de hostname	p 4-5
A- Qu'est-ce qu'un hostname ?	
B- Configuration de hostname	
3-Configuration de l'adresse IP	p 6-8
A- Qu'est-ce qu'un adresse IP?	
B- Les adresse IP statique et dynamique	
C- Configuration de l'adresse IP	
4-Installation du serveur DNS	p 9-18
A-La version du serveur	
B- Installation du paquet bind9	
C- Accéder au fichier resolv.conf	
D- Les fichiers de configuration	
E- Configuration de Bind	
F- Configuration de la clé RNDC	
5-Autorisation d'installation et redémarrage du service bind9	p 19-20
A-Verifier la syntaxe du fichier de configuration DNS	
B-Modifier des propriétaires etdes groupes de fichiers	
C-Redémarrage et Vérification de l'état du service bind9	
6-Tester service bind9	p 21-23

Postfix

1 - Introduction	p 24
A- Qu'est-ce que c'est Postfix ?	
B- Utilisation de Postfix	
2 - Installation de Postfix	p 24-26
A- Obtenir notre hostname	
B- Mettre à Jour les Paquets du Système	
C- Installation de Postfix	
3 - Configuration Postfix	p 27-30
A- Fichier de configuration postfix	
B- Redémarrage et Vérification de l'état du service Postfix	
4 - Crédit d'Email pour l'Envoi	p 30-32
5- Teste du serveur SMTP	p 33-34

Nginx vs Apache

1- Qu'est-ce qu' Un serveur web ?	p 35
2- NGINX vs Apache – Aperçu général	p 35
3- Nginx vs Apache	p 36-37
A-Architecture de base – Gestion des connexions	
B- Comparaison des performances	
C-Sécurité avec Apache et NGINX	
D- Choisir un serveur Web	
3 - Apache et NGINX peuvent-ils fonctionner ensemble	p 38
4 - Conclusion	p 38

Nginx

1-Introduction	p 39
2-Installation de Nginx	p 39-41
A- Mettre à Jour les Paquets du Système	
B- Installation de Nginx	
C- Les processus gèrent et prennent en charge les nouvelles configurations	
D- pages d'accueil par défaut de NGINX	
3-Configuration de Nginx	p 42-49
A- Configuration du Pare-feu UFW	
B- Les fichiers de configuration	
C- Configuration du serveur	
D- Partie http (Port 80)	
E- Partie https (Port 443)	
F- Utilisation de la commande ss -ltn pour lister les connexions TCP en écoute	
G- Autorisation du Trafic HTTPS sur le Port 443 avec UFW	
4- Test Initial du Site Web	p 50-51
5- Configuration finale	p 52-54
A- Accéder au Répertoire du Projet	
B- Installation des Dépendances	
C- Construction du Projet	
D- Copie du répertoire de distribution	
E- Vérification du contenu copié	
F- Modification de la configuration Nginx	
G- Redémarrage de Nginx	
H- Affichage	

Docker

A - Introduction	p 55
B - Qu'est ce que c'est Docker?	p 56
C - Docker vs Machines virtuelles	p 57
D- Comment fonctionne Docker?	p 58-62
D.1 - Image Docker	
D.2 - Conteneur Docker	
D.3 - Docker Registry	
D.4 - Docker Client	
D.5 - Docker Daemon	
D.6 - Architecture de Docker	
D.7 - Docker Compose	
D.8 - Dockerfile	
E - Commandes de base	p 63-65
F - Mise en Pratique : Installation et Configuration de Docker	p 65-85
F.1 - Installer Docker	
F.2 - Créer un compte sur Docker Hub	
F.3 - Installer Docker Desktop	
F.4 - Créer la structure du projet	
F.5 - Mise en Place de l'Environnement de Développement avec Docker pour Laravel	
F.6 - Mise en Place de l'Environnement de Développement avec Docker pour VueJs et Nginx	
****Comparaison entre l'hébergement de Vue.js et Nginx dans un conteneur unique et dans des conteneurs séparés	
F.7 - Configuration de Docker Compose	
F.8 - Gestion des Images Docker	
F.9 - Automatisation avec GitHub	

DNS

1-Introduction

A- Qu'est-ce qu'un DNS ?

Le Domain Name System (Système de nom de domaine) ou DNS est un service informatique distribué qui associe les noms de domaine Internet avec leurs adresses IP ou d'autres types d'enregistrements.



Les internautes accèdent aux informations en ligne via des noms de domaine (par exemple, enfants.innova.ma), tandis que les navigateurs interagissent par le biais d'adresses IP (protocole Internet). Le DNS traduit les noms de domaine en adresses IP afin que les navigateurs puissent charger les ressources web.

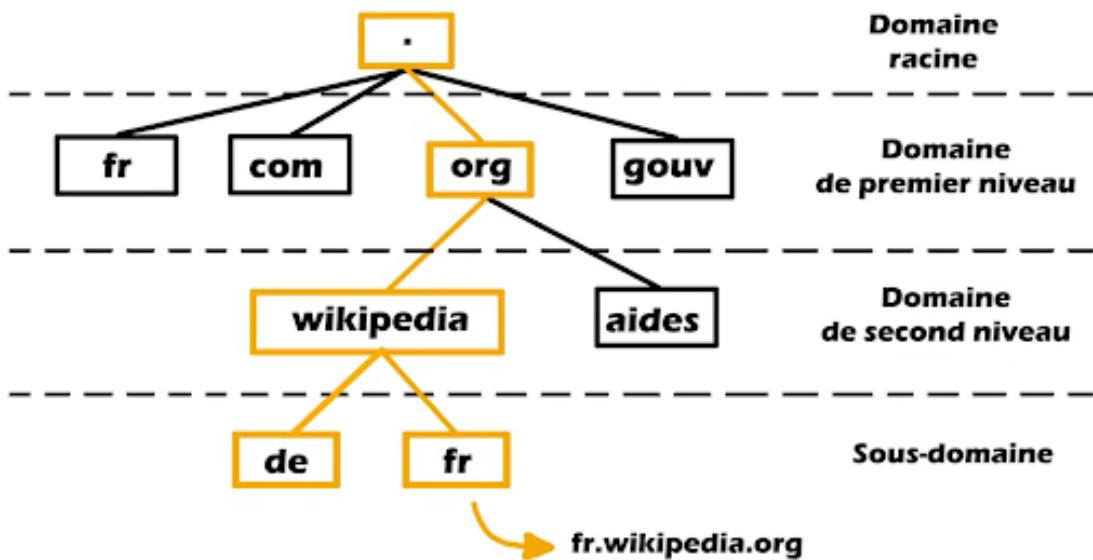
B- Rôle du DNS

Le DNS joue un rôle fondamental dans le fonctionnement d'Internet en facilitant la traduction des noms de domaine en adresses IP et vice versa. Cette traduction permet aux utilisateurs d'accéder aux ressources en ligne en utilisant des noms de domaine conviviaux plutôt que des adresses IP numériques. De plus, le DNS peut être utilisé pour distribuer la charge entre différents serveurs en fournissant plusieurs adresses IP pour le même nom de domaine, ce qui permet d'optimiser les performances et la disponibilité des services.

C- La hiérarchie DNS

L'architecture DNS se compose d'un système de résolution de nom hiérarchique et décentralisé pour les ordinateurs, les services ou toute autre ressource connectée à Internet ou à un réseau privé. Il stocke les différentes informations associées des noms de domaine attribués à chacune des ressources.

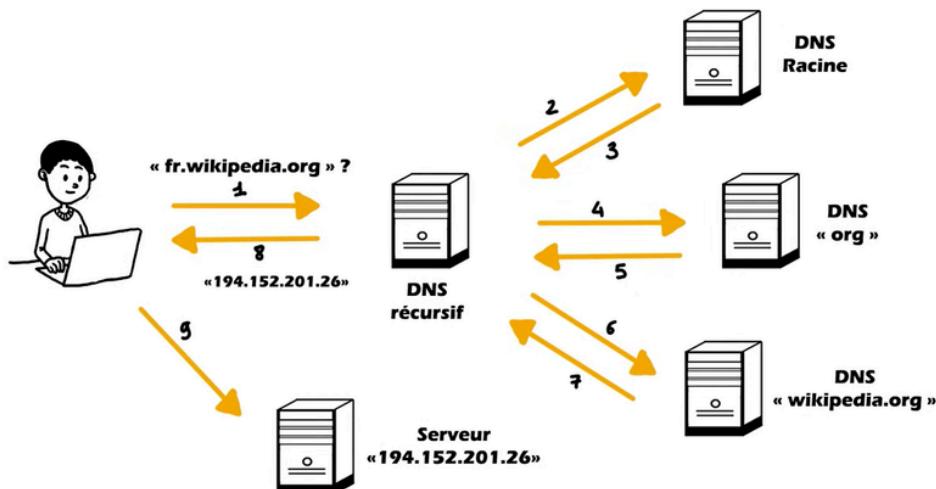
La hiérarchie DNS repose sur plusieurs niveaux qui peuvent intervenir lors d'une résolution DNS :



- **Les serveurs DNS racines**: stocke les informations de la zone racine qui contient tous les noms et adresses IP de tous les domaines de niveau supérieur (TLD).
- **Les serveurs DNS de domaine de premier niveau (TLD)** : Ces serveurs DNS stockent les informations d'enregistrement des domaines de premier niveau. Il s'agit des terminaisons des domaines .com, .net, .fr, .biz, .tv, etc.
- **Les serveurs DNS d'autorité** : Les DNS d'autorités font autorité sur un nom de domaine internet. Ce sont les noms de domaine que les utilisateurs achètent pour leurs sites web, tels que "google.com", "wikipedia.org", etc.
- **DNS récursifs ou itératifs** : Ce sont les serveurs DNS configurés dans la connexion internet de l'appareil et qui sont utilisés lorsqu'un client souhaite effectuer des résolutions DNS. tels que "blog.google.com" ou "fr.wikipedia.org"

Processus de Résolution DNS

Lorsqu'un serveur DNS doit résoudre un nom de domaine pour lequel il ne fait pas autorité, le processus implique plusieurs étapes (généralement de 8 étapes) pour finalement obtenir l'adresse IP associée au nom de domaine. L'exemple ci-dessous décrit l'ensemble des 8 étapes en l'absence d'informations mises en cache:



1
Requête Initiale:
L'utilisateur demande l'adresse IP de fr.wikipedia.org via un serveur DNS récursif.

2
Le DNS récursif demande au DNS racine l'adresse des serveurs DNS responsables du domaine de premier niveau org.

3
Réponse du DNS Racine : Le DNS racine répond avec l'adresse des serveurs DNS pour le domaine org.

4
Le DNS récursif demande au DNS du domaine org l'adresse des serveurs DNS responsables de wikipedia.org.

5
Réponse du DNS org : Le DNS org fournit l'adresse des serveurs DNS pour wikipedia.org.

6
Requête au DNS wikipedia.org : Le serveur DNS récursif demande au DNS de wikipedia.org l'adresse IP pour le sous-domaine fr.

7
Réponse du DNS wikipedia.org: Le DNS wikipedia.org répond avec l'adresse IP 194.152.201.26 pour fr.wikipedia.org.

8
Réponse au Client: Le serveur DNS récursif renvoie l'adresse IP 194.152.201.26 à l'utilisateur.

2-Configuration de hostname

A- Qu'est-ce qu'un hostname ?

Dans Linux, le hostname est le nom unique donné à un ordinateur sur un réseau. Il est utilisé pour identifier cet ordinateur parmi d'autres. Le hostname est généralement assigné lors de l'installation du système d'exploitation ou peut être configuré manuellement par l'utilisateur. Il peut être utilisé pour accéder à cet ordinateur sur le réseau local ou sur Internet.

B- Configuration de hostname

- **Temporaire** : on utilise la commande hostname suivie du nouveau nom d'hôte.

```
$ sudo hostname new-hostname
```

Cette commande changera temporairement le nom d'hôte jusqu'au prochain redémarrage du système. On remplace 'new-hostname' par le nom que nous souhaitons définir comme nom d'hôte temporaire. Après avoir exécuté la commande, on peut vérifier le nouveau nom d'hôte en tapant la commande suivante :

```
$ hostname
```

La sortie affichera le nouveau nom d'hôte temporaire

```
firdaous@innokids-dns:~$ hostname  
innokids-dns  
firdaous@innokids-dns:~$ sudo hostname enfants.innova.ma  
firdaous@innokids-dns:~$ hostname  
enfants.innova.ma  
firdaous@innokids-dns:~$
```

- **Permanente** : Éditez le fichier **/etc/hostname** avec un éditeur de texte (tel que nano ou vim) et remplacez le nom d'hôte actuel par le nouveau. Ce changement deviendra permanent après le redémarrage du système.

```
$ sudo nano /etc/hostname
```

```
GNU nano 6.2                               /etc/hostname
enfants.innova.ma
```

- **Optionnel** : Vous pouvez également associer le nouveau nom d'hôte à l'adresse IP de votre système dans le fichier **/etc/hosts**

```
$ sudo nano /etc/hosts
```

```
GNU nano 6.2                               /etc/hosts *
127.0.0.1      localhost
127.0.1.1      firdaous-HP-EliteBook-630-13-inch-G9-Notebook-PC
192.168.1.5    enfants.innova.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
```

Cette étape est facultative mais peut être utile pour résoudre le nouveau nom d'hôte localement sur le système.

3-Configuration de l'adresse IP

A- Qu'est-ce qu'un adresse IP?

Une adresse IP (Internet Protocol) est un identifiant numérique unique attribué à chaque appareil connecté à un réseau informatique utilisant le protocole Internet pour la communication. Les adresses IP sont utilisées pour localiser et identifier les appareils sur un réseau, leur permettant de communiquer entre eux via Internet ou d'autres réseaux informatiques.

B- Les adresse IP statique et dynamique

Les adresses IP peuvent être attribuées de manière statique ou dynamique en fonction de la manière dont elles sont configurées sur un appareil réseau.

- Une adresse IP statique est une adresse qui est configurée manuellement sur un appareil réseau. cette adresse reste fixe tant que l'administrateur ne la modifie pas manuellement. Les adresses IP statiques sont souvent utilisées pour des équipements réseau critiques tels que des serveurs, des routeurs ou des imprimantes.
- une adresse IP dynamique est une adresse attribuée automatiquement à un appareil par un serveur DHCP (Dynamic Host Configuration Protocol). Les adresses IP dynamiques sont généralement temporaires et peuvent changer à chaque nouvelle connexion au réseau. Elles sont souvent utilisées pour les ordinateurs de bureau, les appareils mobiles et d'autres périphériques qui se connectent fréquemment à différents réseaux.

C- Configuration de l'adresse IP

On utilise généralement ifconfig pour configurer l'adresse IP et le masque de sous-réseau d'un ordinateur, pour afficher des informations sur les interfaces réseau, et pour activer ou désactiver une interface réseau à partir de l'interpréteur de commandes.

```
ftrdaous@innokids-dns:/$ ifconfig  
Command 'ifconfig' not found, but can be installed with:  
sudo apt install net-tools
```

pour utiliser la commande "ifconfig", il est nécessaire d'installer le paquet "net-tools"

```
$ sudo apt update  
$ sudo apt install net-tools
```

- sudo apt update : met à jour la liste des paquets disponibles dans les dépôts configurés sur votre système.

```
ftrdaous@innokids-dns:~$ sudo apt update  
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
113 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

- Les "net-tools" sont un ensemble d'outils réseau qui fournissent des fonctionnalités pour la configuration et la gestion des interfaces réseau sur les systèmes Linux. Ils comprennent des commandes telles que "ifconfig", "netstat", "route", "arp", etc.

```
ftrdaous@innokids-dns:~$ sudo apt install net-tools  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following NEW packages will be installed:  
  net-tools  
0 upgraded, 1 newly installed, 0 to remove and 113 not upgraded.  
Need to get 204 kB of archives.  
After this operation, 819 kB of additional disk space will be used.  
Get:1 http://ma.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]  
Fetched 204 kB in 1s (165 kB/s)  
Selecting previously unselected package net-tools.  
(Reading database ... 162869 files and directories currently installed.)  
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...  
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...  
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...  
Processing triggers for man-db (2.10.2-1) ...
```

Une fois l'installation terminée, on peut utiliser la commande "ifconfig" normalement pour afficher les interfaces réseau de notre système.

```
firdaous@innokids-dns:~$ ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether e0:73:e7:c0:bc:03 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
      device interrupt 16 memory 0x82300000-82320000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 2631 bytes 234495 (234.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2631 bytes 234495 (234.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.107 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::3113:60c:e0d9:7620 prefixlen 64 scopeid 0x20<link>
          ether ac:19:8e:9b:ad:75 txqueuelen 1000 (Ethernet)
            RX packets 69666 bytes 93726274 (93.7 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 38619 bytes 4472735 (4.4 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

pour configurer une adresse IP statique sur l'interface réseau "enp0s31f6" avec l'adresse "192.168.1.5" on utilise la commande :

```
$ sudo ifconfig enp0s31f6 192.168.1.5
$ ifconfig
```

```
firdaous@innokids-dns:~$ sudo ifconfig enp0s31f6 192.168.1.5
firdaous@innokids-dns:~$ ifconfig
enp0s31f6: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
      ether e0:73:e7:c0:bc:03 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
      device interrupt 16 memory 0x82300000-82320000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 2727 bytes 245654 (245.6 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 2727 bytes 245654 (245.6 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp0s20f3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.1.107 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::3113:60c:e0d9:7620 prefixlen 64 scopeid 0x20<link>
          ether ac:19:8e:9b:ad:75 txqueuelen 1000 (Ethernet)
            RX packets 70604 bytes 94429048 (94.4 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 39414 bytes 4686966 (4.6 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Pour que les modifications prennent effet et activer manuellement l'interface il faut soit redémarrer l'interface réseau, soit utiliser la commande:

```
$ sudo ifconfig enp0s31f6 up
```

4-Installation du serveur DNS

A-La version du serveur

la version du serveur est la spécification de la version du logiciel serveur utilisé pour fournir des services sur un réseau, et elle est cruciale pour assurer la compatibilité, la sécurité et l'efficacité des opérations du serveur.

La commande pour connaître la version du serveur sur Linux :

```
$ lsb_release -a && ip r
```

- La commande **lsb_release -a** affiche des informations sur la distribution Linux installée sur notre système. Dans ce cas, il indique que nous utilisons Ubuntu version 22.04.4 LTS, également connue sous le nom de "jammy".
- La commande **ip r** affiche la table de routage du système. Dans ce cas, il montre qu'il y a une route vers le réseau 192.168.1.0/24 via l'interface réseau "enp0s31f6" avec une adresse source de 192.168.1.5. Cependant, il indique également que cette interface est "linkdown", ce qui signifie qu'elle n'est pas actuellement en ligne ou qu'elle est désactivée.

```
firdaous@innoktds-dns:~$ lsb_release -a && ip r
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.4 LTS
Release:        22.04
Codename:       jammy
192.168.1.0/24 dev enp0s31f6 proto kernel scope link src 192.168.1.5 linkdown
```

B- Installation du paquet bind9

BIND9 (Berkeley Internet Name Domain) est une implémentation open source largement utilisée du serveur DNS, développée par l'Internet Systems Consortium (ISC). En tant que serveur DNS, BIND9 est responsable de la résolution des noms de domaine en adresses IP et vice versa, facilitant ainsi la communication entre les dispositifs connectés sur Internet.

Commençons par installer le paquet bind9 pour transformer notre serveur en serveur DNS, grâce aux commandes suivantes :

```
$ sudo apt install bind9
```

```
firdaous@innokids-dns:~$ sudo apt install bind9
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bind9-utils
Suggested packages:
  bind-doc resolvconf
The following NEW packages will be installed:
  bind9 bind9-utils
0 upgraded, 2 newly installed, 0 to remove and 113 not upgraded.
```

C- Accéder au fichier resolv.conf

Le fichier **resolv.conf** sur Linux configure la résolution DNS en définissant les serveurs DNS à utiliser et les options de recherche de domaine. Cela permet la conversion des noms de domaine en adresses IP, essentielle pour la connectivité Internet.

Pour accéder au fichier resolv.conf:

```
$ sudo nano /etc/resolv.conf
```

Dans ce fichier, nous ajoutons les éléments suivants :

- **nameserver 192.168.1.5**
- **nameserver 8.8.8.8**
- **options edns0 trust-ad**
- **search enfants.innova.ma"**

```

GNU nano 6.2                               /etc/resolv.conf
This is /run/systemd/resolve/stub-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 to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# 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
options edns0 trust-ad
search enfants.innova.ma

```

Nameservers :

- **nameserver 192.168.1.5** : Spécifie le premier serveur DNS à utiliser, dans ce cas l'adresse IP est **192.168.1.5**.
- **nameserver 8.8.8.8** : Spécifie le deuxième serveur DNS à utiliser, ici l'adresse IP est **8.8.8.8**. Ce serveur DNS est fourni par Google Public DNS.

Options :

- **options edns0 trust-ad** : Configure les options DNS. Dans ce cas, cela active EDNS0 (Extension Mechanisms for DNS) et fait confiance aux réponses AD (Authenticated Data) pour les données DNS sécurisées.

Search :

- **search enfants.innova.ma** : Définit le domaine de recherche par défaut. Cela signifie que si un nom de domaine n'est pas entièrement qualifié (c'est-à-dire s'il ne se termine pas par un point), le système ajoutera automatiquement ce domaine de recherche lorsqu'il tentera de le résoudre.

NB: le fichier **resolv.conf**, qui spécifie les serveurs DNS à utiliser, peut être régénéré automatiquement par des processus de gestion du réseau à chaque redémarrage ou reconfiguration du réseau. Dans certains cas, les modifications apportées directement à ce fichier peuvent être écrasées. Il est donc important de noter que vous devrez potentiellement reconfigurer ce fichier à chaque fois que vous souhaitez définir ou modifier les serveurs DNS.

D- Les fichiers de configuration

Les fichiers de configuration de Bind se trouvent dans le répertoire suivant :

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

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

Ce répertoire contient les fichiers de configuration suivant :

- Les fichiers "db." correspondent aux fichiers de zones incluent par défaut dans **Bind**. Ils vont nous servir de modèle pour la création de nos fichiers de zones.
- Le fichier "**named.conf**" est le fichier de configuration de Bind9, dans lequel on trouve un lien vers 3 autres fichiers : "**named.conf.options**" contient les options de configuration de Bind; "**named.conf.local**" sert à déclarer des zones; "**named.conf.default-zones**" contient la définition des zones incluses par défaut avec Bind.

E- Configuration de Bind

Le fichier de configuration global de Bind est le fichier "**named.conf.options**".

```
$ sudo nano named.conf.options
```

```
GNU nano 6.2                                              named.conf.options  
options {  
    directory "/var/cache/bind";  
  
    // If there is a firewall between you and nameservers you want  
    // to talk to, you may need to fix the firewall to allow multiple  
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113  
  
    // If your ISP provided one or more IP addresses for stable  
    // nameservers, you probably want to use them as forwarders.  
    // Uncomment the following block, and insert the addresses replacing  
    // the all-0's placeholder.  
  
    // forwarders {  
    //     0.0.0.0;  
    // };  
  
    //=====  
    // If BIND logs error messages about the root key being expired,  
    // you will need to update your keys. See https://www.isc.org/bind-keys  
    //=====  
    dnssec-validation auto;  
  
};  
listen-on-v6 { any; };
```

On peut voir qu'il y a beaucoup de lignes commentées (celles qui débutent par "//"), en ce qui concerne les options suivants:

- **directory** : il s'agit du chemin vers le répertoire qui contient les fichiers de zone.
- **dnssec-validation auto** : la valeur par défaut est yes, le **DNSSEC** est un système qui permet de sécuriser les échanges **DNS**.
- **listen-on-v6** : Active le fait de résoudre les requêtes sur des adresses **IPv6**.

Nous avons configuré notre serveur BIND pour qu'il utilise les serveurs DNS de Google (8.8.8.8 et 8.8.4.4) comme forwarders. Cela signifie que lorsque notre serveur DNS BIND reçoit une requête pour un nom de domaine qu'il ne peut pas résoudre à partir de ses propres enregistrements, il transmettra cette requête aux serveurs DNS de Google pour obtenir la réponse. Voici la modification dans le fichier named.conf.options :

```
GNU nano 6.2                                              named.conf.options
options {
    directory "/var/cache/bind";

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

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

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

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    =====
    // 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; };
};
```

Ouvrons le fichier named.conf.default-zone avec un éditeur de texte.
Nous pouvons utiliser la commande suivante :

```
$ sudo nano named.conf.default-zone
```

```

GNU nano 6.2                               named.conf.default-zones
// prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "/etc/bind/db.255";
};

```

Ouvrons le fichier named.conf.local avec la commande suivante:

```
$ sudo nano named.conf.local
```

Copions les configurations de La zone "localhost" et de la zone "127.in-addr.arpa" de fichier named.conf.default-zone et collons-les à la fin du fichier naed.conf.local :

```

GNU nano 6.2                               named.conf.local *
//                                         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 "localhost" {
    type master;
    file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "/etc/bind/db.127";
};

```

La présence des zones "localhost" et "127.in-addr.arpa" dans le fichier named.conf.local est une pratique courante dans la configuration de BIND. Ces zones sont des zones essentielles pour la résolution des noms d'hôtes et des adresses IP de bouclage, et elles sont souvent incluses dans ce fichier de configuration pour une gestion plus modulaire.

Le fichier named.conf.local est généralement utilisé pour inclure des configurations spécifiques à notre environnement local. Ces configurations peuvent inclure des zones supplémentaires, des options de transfert de zone et des configurations de sécurité.

Dans notre fichier named.conf.local, nous allons modifi   les deux nouvelles zones :

- **La zone enfants.innova.ma** est une zone directe (type master) avec le fichier de donn  es /etc/bind/db.direct, qui mappe les noms de domaine    des adresses IP.
- **La zone 1.168.192.in-addr.arpa** est une zone de r  solution inverse (type master) avec le fichier de donn  es /etc/bind/db.inverse, qui mappe les adresses IP du r  seau 192.168.1.0/24    des noms de domaine. Nous utilisons 1.168.192.in-addr.arpa plut  t que 192.168.1 car le format de r  solution inverse n  cessite d'inverser l'ordre des octets et d'ajouter le suffixe in-addr.arpa.

```
| GNU nano 6.2                                     named.conf.local
// Do any local configuration here
//
zone "enfants.innova.ma" IN{
    type master;
    file "/etc/bind/db.direct";
};

zone "1.168.192.in-addr.arpa" IN{
    type master;
    file "/etc/bind/db.inverse";
};

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

on va cr  er et configurer les fichier (db.direct et db.inverse) dans le r  pertoire /etc/bind.

```
$ sudo touch db.direct
$ sudo touch db.inverse
```

```
firdaous@innokids-dns:/etc/bind$ sudo touch db.direct
[sudo] password for firdaous:
firdaous@innokids-dns:/etc/bind$ sudo touch db.inverse
firdaous@innokids-dns:/etc/bind$ ls
bind.keys  db.255   db.inverse  named.conf.default-zones  rndc.key
db.0       db.direct  db.local   named.conf.local        zones.rfc1918
db.127    db.empty   named.conf  named.conf.options
firdaous@innokids-dns:/etc/bind$
```

et on va copier le fichier `db.local` vers `db.direct` avec la commande:

```
$ sudo cp db.local db.direct
```

On va ouvrir le fichier db.direct :

```
$ sudo nano db.direct
```

```
GNU nano 6.2                               db.direct
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                        2           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800      ; Negative Cache TTL
;
@       IN      NS      localhost.
@       IN      A       127.0.0.1
@       IN      AAAA   ::1
```

Ce fichier de configuration BIND est utilisé pour définir la résolution DNS pour le domaine "enfants.innova.ma", spécifiant que le serveur DNS responsable de ce domaine est ns.enfants.innova.ma et que les noms de domaine "ns" et "www" sont tous deux associés à l'adresse IP 192.168.1.5.

```
GNU nano 6.2                               db.direct *
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns.enfants.innova.ma. root.ns.enfants.innova.ma. (
                        2           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800      ; Negative Cache TTL
;
;nameserver -NS records
        IN      NS     ns.enfants.innova.ma.

;nameserver -A records
ns.enfants.innova.ma.  IN      A      192.168.1.5
www.enfants.innova.ma. IN      A      192.168.1.5
;
```

Ce fichier indique que les noms de domaine "ns.enfants.innova.ma" et "www.enfants.innova.ma" sont tous deux associés à l'adresse IP 192.168.1.5 à l'aide d'enregistrements de type A (adresses IPv4). Cela signifie que lorsque quelqu'un effectue une recherche de résolution inverse pour l'adresse IP 192.168.1.5,

et on va copié le fichier `db.127` vers `db.inverse` e avec la commande:

```
$ sudo cp db.127 db.inverse
```

On va ouvrir le fichier db.inverse :

```
$ sudo nano db.direct
```

```
GNU nano 6.2                                     db.inverse
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA    localhost. root.localhost. (
                        1           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )     ; Negative Cache TTL
;
@       IN      NS     localhost.
1.0.0  IN      PTR    localhost.
```

Ce fichier de configuration BIND est crucial pour la gestion de la résolution DNS inversée pour le domaine "enfants.innova.ma".

```
GNU nano 6.2                                     db.inverse *
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA    ns.enfants.innova.ma. root.ns.enfants.innova.ma. (
                        1           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )     ; Negative Cache TTL
;
@       IN      NS     ns.enfants.innova.ma.
5       IN      PTR    ns.enfants.innova.ma.
5       IN      PTR    www.enfants.innova.ma.
```

Les enregistrements PTR (@ IN PTR) jouent un rôle essentiel ici en associant l'adresse IP 192.168.1.5 aux noms de domaine "ns.enfants.innova.ma" et "www.enfants.innova.ma". Contrairement à la résolution DNS directe où un nom de domaine est associé à une adresse IP, la résolution inverse permet de déterminer le nom de domaine à partir d'une adresse IP donnée. Ainsi, lorsqu'une recherche de résolution inverse est effectuée pour l'adresse IP 192.168.1.5, le système renvoie ces noms de domaine spécifiques, facilitant l'identification et la gestion du réseau pour les serveurs et les services associés à cette adresse IP.

F- Configuration de la clé RNDC:

RNDC est un outil utilisé pour contrôler et administrer un serveur BIND à distance.

```
firdaous@innokids-dns:/etc/bind$ sudo cat rndc.key
key "rndc-key" {
        algorithm hmac-sha256;
        secret "q07Dt7AuYEz2XBfRiQYgUkW9aFdosB/gVGFBMjQ+kM=";
};
```

cette configuration permet d'établir une communication sécurisée entre le serveur BIND et l'outil RNDC, garantissant que seules les commandes signées avec la clé secrète spécifiée seront acceptées et exécutées par le serveur BIND. Cela renforce la sécurité globale du système DNS en empêchant les commandes non autorisées ou malveillantes d'être traitées par le serveur BIND.

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

```
GNU nano 6.2                                     rndc.conf *
key "rndc-key" {
    algorithm hmac-sha256;
    secret "q07Dt7AuYEz2XBfRiQYgUkW9aFdosB/gVGFBMjQ+kM=";
};

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

Voici une explication détaillée des éléments mentionnés dans la configuration RNDC (Remote Name Daemon Control) pour le serveur BIND :

options {...} : Cette section configure les options RNDC spécifiques pour définir comment RNDC interagit avec le serveur BIND.

- **default-key "rndc-key";** : Spécifie que la clé par défaut à utiliser pour toutes les commandes RNDC envoyées au serveur BIND est "rndc-key".
- **default-server 127.0.0.1;** : Définit l'adresse IP du serveur BIND. il est configuré pour utiliser le serveur BIND fonctionnant sur la même machine, car l'adresse IP est localhost (127.0.0.1).
- **default-port 953;** : Spécifie le port par défaut sur lequel le serveur BIND écoute les commandes RNDC. Le port 953 est conventionnellement réservé pour les commandes de contrôle RNDC dans BIND.

Ce fichier de configuration permet une gestion centralisée et sécurisée du serveur BIND à distance en utilisant l'outil RNDC. En utilisant une clé RNDC forte, spécifiant le serveur BIND et le port par défaut, il assure une administration efficace tout en maintenant la sécurité et l'intégrité du système DNS.

5-Autorisation d'installation et redémarrage du service bind9

A-Verifier la syntaxe du fichier de configuration DNS:

On peut vérifier la syntaxe du fichier de configuration named.conf et des fichiers de zone inclus avec la commande:

```
$ named-checkconf -z
```

La commande `named-checkzone` est utilisée aussi pour vérifier la syntaxe d'un seul fichier de zone DNS.

Dans notre cas, nous avons utilisé la commande `named-checkzone` pour vérifier le fichier de zone pour le domaine `enfants.innova.ma` en spécifiant le chemin du fichier `/etc/bind/db.direct`

```
$ named-checkconf enfants.innova.ma /etc/bind/db.direct
```

```
firdaous@innokids-dns:/etc/bind$ named-checkconf -z
zone enfants.innova.ma/IN: loaded serial 2
zone 1.168.192.in-addr.arpa/IN: loaded serial 2
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
firdaous@innokids-dns:/etc/bind$ named-checkzone enfants.innova.ma /etc/bind/db.direct
zone enfants.innova.ma/IN: loaded serial 2
OK
```

B-Modifier des propriétaires et des groupes de fichiers:

```
$ sudo chown root:bind -R /etc/bind
```

Cette commande modifie récursivement les propriétaires et les groupes de tous les fichiers et répertoires dans /etc/bind et ses sous-répertoires. Cela attribue l'utilisateur **root** comme propriétaire et le groupe **bind** comme groupe pour ces fichiers. Cette mesure renforce la sécurité en restreignant l'accès aux fichiers de configuration de BIND aux utilisateurs autorisés, limitant ainsi les modifications non autorisées.

```
$ sudo chown bind:bind -R /etc/bind/rndc.conf
```

Cette commande spécifie que le fichier rndc.conf situé dans le répertoire /etc/bind appartient à l'utilisateur bind et au groupe bind. Cette action est souvent entreprise pour sécuriser le fichier de configuration RNDC, assurant que seuls les processus liés à BIND, généralement exécutés sous l'utilisateur bind, peuvent y accéder.

C-Redémarrage et Vérification de l'état du service bind9:

```
$ sudo service bind9 restart
```

Cette commande redémarre le service BIND9 sur notre système. Cela arrête d'abord le service s'il est en cours d'exécution, puis le redémarre, appliquant ainsi les éventuels changements de configuration.

```
$ sudo service bind9 status
```

Cette commande est utilisée pour vérifier l'état actuel du service BIND9 . Elle affiche des informations sur le fonctionnement du service, indiquant s'il est en cours d'exécution, arrêté, ou s'il rencontre des problèmes. C'est un outil essentiel pour diagnostiquer et résoudre les problèmes liés au service DNS sur un système Linux.

```
firdaous@innokids-dns:/etc/bind$ sudo chown root:bind -R /etc/bind
firdaous@innokids-dns:/etc/bind$ sudo chown bind:bind -R /etc/bind/rndc.conf
firdaous@innokids-dns:/etc/bind$ sudo service bind9 restart
firdaous@innokids-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 Fri 2024-05-03 00:47:45 +01; 41s ago
       Docs: man:named(8)
    Process: 7156 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
      Main PID: 7157 (named)
         Tasks: 38 (limit: 28308)
        Memory: 9.1M
          CPU: 81ms
        CGroup: /system.slice/named.service
                   └─7157 /usr/sbin/named -u bind

00:47:45 03 مـ enfants.innova.ma systemd[1]: Started BIND Domain Name Server.
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:dc3::35>
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:500:2::>
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:503:ba3>
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:500:1::>
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:7fd::1#>
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:500:9f:>
00:47:45 03 مـ enfants.innova.ma named[7157]: network unreachable resolving './NS/IN': 2001:500:2d:>
00:47:45 03 مـ enfants.innova.ma named[7157]: managed-keys-zone: Key 20326 for zone . is now trustee>
00:47:45 03 مـ enfants.innova.ma named[7157]: resolver priming query complete: success
```

6-Tester service bind9

La sortie de la commande dig ns.enfants.innova.ma montre que la résolution DNS pour le sous-domaine "ns.enfants.innova.ma" a été effectuée avec succès.

```
$ dig ns.enfants.innova.ma
```

```
firdaous@innokids-dns:~$ dig ns.enfants.innova.ma

; <>> DIG 9.18.18-0Ubuntu0.22.04.2-Ubuntu <>> ns.enfants.innova.ma
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39020
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: a122ceb436372d1601000000663428c7517cf9af5a27bf85 (good)
;; QUESTION SECTION:
;ns.enfants.innova.ma.      IN      A

;; ANSWER SECTION:
ns.enfants.innova.ma.  604800  IN      A      192.168.1.5

;; Query time: 0 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Fri May 03 00:59:03 +01 2024
;; MSG SIZE rcvd: 93
```

La sortie de la commande dig www.enfants.innova.ma indique que la résolution DNS pour le sous-domaine "www.enfants.innova.ma" a été effectuée avec succès.

```
$ dig www.enfants.innova.ma
```

```
firdaous@innokids-dns:~$ dig www.enfants.innova.ma

; <>> DIG 9.18.18-0Ubuntu0.22.04.2-Ubuntu <>> www.enfants.innova.ma
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33999
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: ccf0245e3e74249201000000663429440a01e570d88ba450 (good)
;; QUESTION SECTION:
;www.enfants.innova.ma.      IN      A

;; ANSWER SECTION:
www.enfants.innova.ma.  604800  IN      A      192.168.1.5

;; Query time: 4 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Fri May 03 01:01:08 +01 2024
;; MSG SIZE rcvd: 94
```

La sortie de la commande dig -x 192.168.1.5 indique une recherche de résolution DNS inverse pour l'adresse IP 192.168.1.5.

```
$ dig -x 192.168.1.5
```

```
firdaous@innokids-dns:~$ dig -x 192.168.1.5

; <>> DiG 9.18.18-0ubuntu0.22.04.2-Ubuntu <>> -x 192.168.1.5
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 863
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 1232
;; COOKIE: 62d8bcc47980d68c0100000066342a1d9b2082652d2ca188 (good)
;; QUESTION SECTION:
;5.1.168.192.in-addr.arpa. IN PTR

;; ANSWER SECTION:
5.1.168.192.in-addr.arpa. 604800 IN PTR ns.enfants.innova.ma.
5.1.168.192.in-addr.arpa. 604800 IN PTR www.enfants.innova.ma.

;; Query time: 3 msec
;; SERVER: 192.168.1.5#53(192.168.1.5) (UDP)
;; WHEN: Fri May 03 01:04:45 +01 2024
;; MSG SIZE rcvd: 133
```

La commande nslookup www effectue une recherche de résolution DNS pour le nom d'hôte "www" en utilisant le serveur DNS configuré sur 192.168.1.5.

```
$ nslookup www
```

```
firdaous@innokids-dns:~$ nslookup www
Server:      192.168.1.5
Address:      192.168.1.5#53

Name:    www.enfants.innova.ma
Address: 192.168.1.5
```

La commande nslookup www.enfants.innova.ma est utilisée pour interroger un serveur DNS afin de récupérer l'adresse IP associée au nom de domaine "www.enfants.innova.ma". Cela peut être utile pour vérifier si la résolution DNS fonctionne correctement pour ce domaine.

```
$ nslookup www.enfants.innova.ma
```

```
firdaous@enfants-innova-ma:~$ nslookup www.enfants.innova.ma
Server:      192.168.1.5
Address:     192.168.1.5#53

Name:   www.enfants.innova.ma
Address: 192.168.1.5
```

La commande nslookup avec une adresse IP comme argument effectue une recherche de résolution inverse pour trouver les noms de domaine associés à cette adresse IP. Dans notre cas, en exécutant nslookup 192.168.1.5, la sortie montre que deux noms de domaine sont associés à l'adresse IP 192.168.1.5 :

- ns.enfants.innova.ma
- www.enfants.innova.ma

Cela signifie que l'adresse IP 192.168.1.5 est associée à ces deux noms de domaine dans la zone inversée 192.168.1.in-addr.arpa, ce qui est cohérent avec les résultats de la commande dig -x 192.168.1.5 que nous avez précédemment exécutée.

```
$ nslookup 192.168.1.5
```

```
firdaous@innokids-dns:~$ nslookup 192.168.1.5
5.1.168.192.in-addr.arpa      name = ns.enfants.innova.ma.
5.1.168.192.in-addr.arpa      name = www.enfants.innova.ma.
```

POSTFIX

1-Introduction

A- Qu'est-ce que c'est Postfix ?

Postfix est un serveur de messagerie électronique et un logiciel libre. Il se charge de la livraison de messages électroniques et a été conçu comme une alternative plus rapide, plus facile à administrer et plus sécurisée que l'historique Sendmail.



B- Utilisation de Postfix

Postfix est souvent utilisé par les administrateurs de serveurs Linux et Unix pour gérer les services de messagerie. Il est couramment intégré dans des environnements d'hébergement Web, des entreprises, et des infrastructures de messagerie à grande échelle. De plus, il peut être couplé avec d'autres logiciels comme Dovecot pour la gestion des boîtes aux lettres, ou SpamAssassin et ClamAV pour le filtrage des spams et des virus.

En somme, Postfix se distingue par sa combinaison de sécurité, de performance, et de flexibilité, en faisant un choix privilégié pour de nombreux administrateurs de systèmes qui cherchent à mettre en place un serveur de messagerie fiable et efficace.

2-Installation de Postfix

A- Obtenir notre hostname :

```
firdaous@innokids-postfix:~$ hostname  
enfants.innova.ma  
firdaous@innokids-postfix:~$ hostname -f  
enfants.innova.ma  
firdaous@innokids-postfix:~$ hostname --fqdn  
enfants.innova.ma
```

B- Mettre à Jour les Paquets du Système :

Pour maintenir notre système à jour avec les dernières versions des logiciels, il est essentiel de rafraîchir les informations sur les paquets disponibles. Cette actualisation de la base de données locale des paquets garantit que nous disposons des informations les plus récentes avant d'installer ou de mettre à jour des logiciels.

```
$ sudo apt update
```

```
firdaous@innokids-postfix:~$ sudo apt update
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:4 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 229 kB in 1s (202 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
114 packages can be upgraded. Run 'apt list --upgradable' to see them.
firdaous@innokids-postfix:~$
```

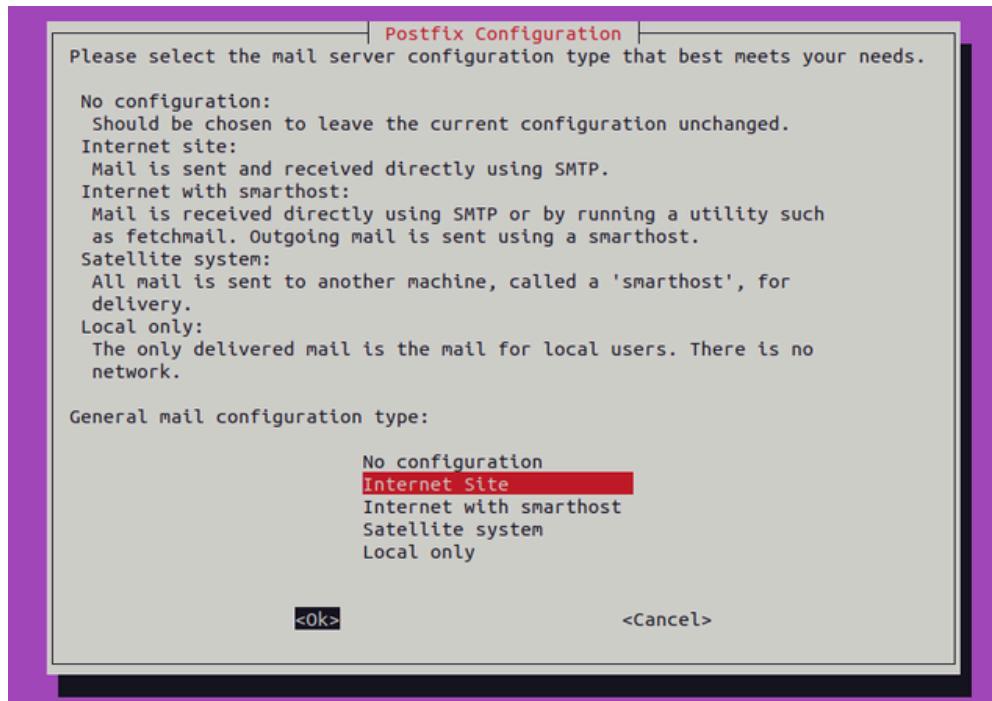
C- Installation de Postfix :

```
$ sudo apt install postfix
```

Après avoir exécuté la commande sudo apt install postfix, la configuration de Postfix s'ouvre. Voici les options principales :

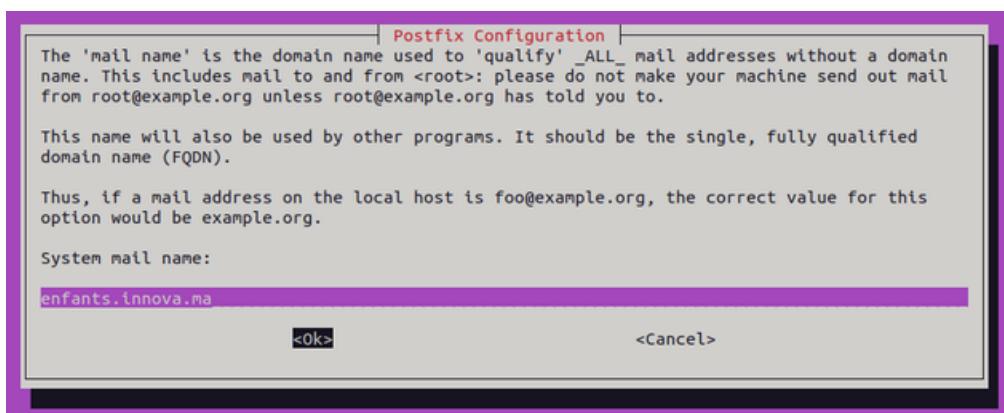
- **Internet Site** : Cette option est utilisée lorsque notre serveur Postfix est destiné à recevoir et à envoyer des e-mails sur Internet. C'est l'option la plus courante pour la plupart des serveurs de messagerie.
- **Satellite system** : Cette option est utilisée lorsque notre serveur Postfix agit comme un satellite qui envoie tous les e-mails à un serveur de messagerie (smarthost) pour traitement.
- **Internet with smarthost** : Cette option est similaire à "Satellite system", mais elle accepte également les e-mails entrants pour notre domaine. Cela signifie que notre serveur Postfix agit comme un relais pour le courrier entrant et sortant, mais utilise un smarthost pour relayer les e-mails sortants.

Dans la plupart des cas, si nous avons un serveur de messagerie destiné à envoyer et recevoir des e-mails sur Internet, l'option "Site Internet" est celle que nous choisirions.



Cette configuration de Postfix nous demande de spécifier le nom de domaine à utiliser pour qualifier toutes les adresses e-mail sans domaine spécifié. Cela inclut les e-mails envoyés à et depuis l'utilisateur root.

Dans notre cas, le nom de domaine actuellement spécifié est "enfants.innova.ma". Cela signifie que toutes les adresses e-mail sans domaine spécifié seront qualifiées avec "enfants.innova.ma".



les fichiers et dossiers présents dans le répertoire de configuration de Postfix:

```
daous@innokids-postfix:~$ ls /etc/postfix
main.cf      nakedefs.out    master.cf.proto   postfix-files.d  post-install
main.cf.d    main.cf.proto   master.cf       postfix-files   postfix-script  sasl
daous@innokids-postfix:~$ 
```

3-configuration Postfix

A- Fichier de configuration postfix

Le fichier de configuration de Postfix est le fichier /etc/postfix/main.cf

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

```
firdaous@innokids-postfix:~$ ls /etc/postfix
dynamicmaps.cf      main.cf          makedefs.out    master.cf.proto  postfix-files.d  post-install
dynamicmaps.cf.d    main.cf.proto   master.cf      postfix-files   postfix-script  sasl
firdaous@innokids-postfix:~$ sudo nano /etc/postfix/main.cf
```

Il est d'abord important de connaitre quelques principaux paramètres:

- **myhostname:** Ce paramètre désigne le nom complet de l'ordinateur, le FQDN du serveur. Si cette option est commentée, postfix va vérifier le résultat de la commande hostname. Il faut vérifier donc /etc/hosts.
- **mydestination:** Ce paramètre inclut le nom complet de la machine ainsi que des synonymes de localhost. Il faut bien vérifier que le (ou les) domaine principale soit renseigné. D'une manière général, il convient habituellement ici de donner tous les noms de domaines pour lesquels cette machine fais office de serveur MX.
- **mynetworks:** indique les réseaux habilités à envoyer du courrier par l'intermédiaire de cette machine. Par défaut Postfix est configuré pour n'accepter que des courriers issus de la machine elle-même. Il faut généralement ajouter le réseau local.
- **home_mailbox:** indique le répertoire de stockage du courrier qui pourra être rapatrier plus tard avec POP ou IMAP.

C'est dans ce fichier que vous configurez la plupart des paramètres de base de votre serveur de messagerie Postfix.

```

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_CApath=/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 = enfants.innova.ma
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, enfants.innova.ma, localhost.innova.ma, , localhost
relayhost =
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

```

Nous avons besoin de personnaliser davantage la configuration de Postfix pour répondre à nos besoins spécifiques, et nous pouvons le faire en modifiant le fichier main.cf.

Les changements que nous avons apportés :

relayhost = [smtp.gmail.com]:587

Cette ligne spécifie le serveur SMTP de Gmail et le port SMTP à utiliser pour l'envoi d'e-mails sortants. [smtp.gmail.com] est l'adresse du serveur SMTP, et 587 est le port SMTP utilisé pour les connexions chiffrées TLS.

```
# 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
```

- **smtp_sasl_auth_enable = yes** : Active l'authentification SASL pour les connexions SMTP sortantes, permettant à notre serveur de s'authentifier auprès du serveur SMTP distant comme celui de Gmail.
- **smtp_sasl_security_options = noanonymous** : Configure SASL pour n'autoriser que des connexions SMTP sortantes authentifiées.
- **smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd** : Spécifie le chemin du fichier sasl_passwd contenant les informations d'identification nécessaires pour l'authentification auprès du serveur SMTP distant.
- **smtp_tls_security_level = encrypt** : Configure Postfix pour utiliser TLS afin de chiffrer les connexions SMTP sortantes, améliorant ainsi la sécurité des e-mails envoyés.
- **smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt** : Indique le chemin du fichier contenant les autorités de certification utilisées pour vérifier les certificats lors des connexions à des serveurs SMTP distants comme ceux de Gmail.

Ces ajustements renforcent la sécurité et l'authentification des e-mails sortants via notre serveur Postfix, garantissant leur envoi de manière sécurisée et fiable.

B- Redémarrage et Vérification de l'état du service Postfix:

```
$ sudo systemctl restart postfix
```

Cette commande redémarre le service Postfix sur notre système. Cela permet de charger les nouvelles configurations que nous avons effectuées dans le fichier main.cf .

```
$ sudo systemctl status postfix
```

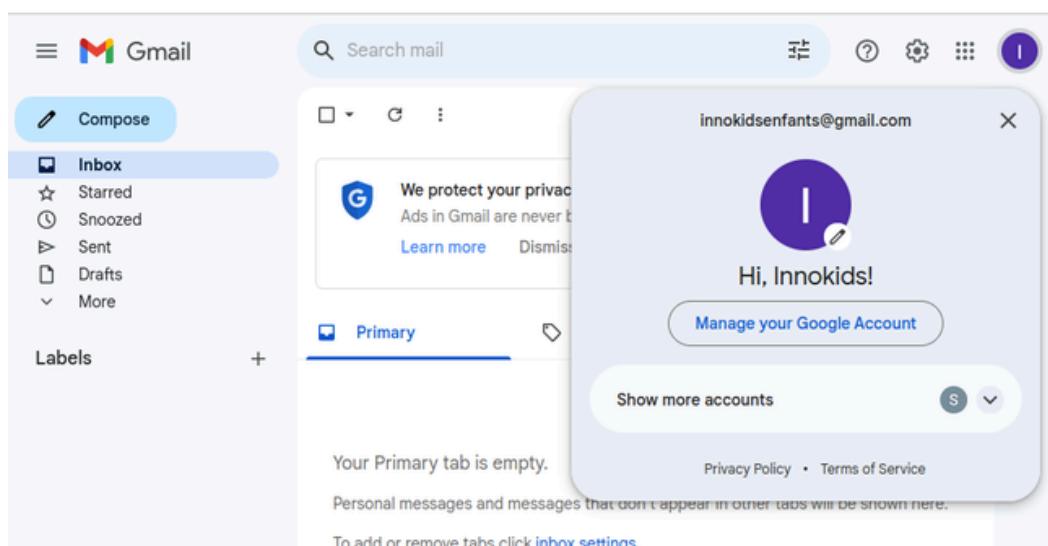
Cette commande affichera le statut actuel du service Postfix sur votre système.

```
firdaous@innokids-postfix:~$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
   Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
   Active: active (exited) since Sun 2024-05-05 23:49:46 +01; 1min 4s ago
     Docs: man:postfix(1)
 Process: 18846 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 18846 (code=exited, status=0/SUCCESS)
    CPU: 1ms

23:49:46 05 ⌁ enfants.innova.ma systemd[1]: Starting Postfix Mail Transport Agent...
23:49:46 05 ⌁ enfants.innova.ma systemd[1]: Finished Postfix Mail Transport Agent.
```

4- Crédit d'Email pour l'Envoi:

D'abord on doit créer un email qui va prendre en charge l'opération d'envoi des emails



Se connecter au compte Google associé à notre adresse email

Home Personal info Data & privacy Security People & sharing Payments & subscriptions

Welcome, Innokids enfants

Manage your info, privacy, and security to make Google work better for you. [Learn more](#)

Privacy & personalization
See the data in your Google Account and choose what activity is saved to personalize your Google experience

[Manage your data & privacy](#)

You have security tips
Security tips found in the Security Checkup

[Review security tips](#)

Activer la validations en deux étape :

TOUR ACCOUNT IS PROTECTED

Home Personal info Data & privacy **Security** People & sharing Payments & subscriptions

mended actions

[See details](#)

Recent security activity

Signing in with 2-Step Verification was turned on	11:54 AM · Morocco	>
Recovery phone added	11:53 AM · Morocco	>
New sign-in on Redmi Note 11S	2:19 AM · Morocco	>

[Review security activity \(7\)](#)

How you sign in to Google

Make sure you can always access your Google Account by keeping this information up to date

2-Step Verification	On since 11:54 AM	>
Passkeys and security keys	2 passkeys	>
Password	Last changed May 3	>
Skip password when possible	On	>

gérer une "Mot de passe d'application":

Lors de la génération du mot de passe d'application, nous pouvons nommer l'application comme "Postfix" ou "Serveur SMTP". Ce nom sert à identifier l'application associée au mot de passe dans votre compte Google, sans lien direct avec le nom du service ou logiciel configuré sur votre système Linux.

The screenshot shows the 'Mots de passe des applications' (Application Passwords) section of the Google Account settings. It includes a descriptive text about application passwords, a note about creating a specific password for an app, and a form where 'mail' is entered as the app name. A 'Créer' (Create) button is visible at the bottom right of the form.

Vous n'avez aucun mot de passe d'application.
Pour créer un mot de passe spécifique à une appli, indiquez son nom ci-dessous.
Nom de l'appli
mail

Créer

Lorsque nous cliquons sur "Créer", Google va générer un mot de passe d'application pour l'application que nous avons spécifiée.

The screenshot shows a modal dialog titled 'Mot de passe d'application généré' (Generated Application Password). It displays the generated password 'lllr moyx ucis kxeh'. Below the password, there is a 'Comment l'utiliser?' (How to use it?) section with instructions on how to use the password for an app. An 'OK' button is at the bottom right of the modal.

Mot de passe d'application généré
Mot de passe d'application pour votre appareil
lllr moyx ucis kxeh
Comment l'utiliser ?
Accédez aux paramètres de votre compte Google dans l'application ou l'appareil que vous essayez de configurer. Remplacez le mot de passe par celui de 16 caractères indiqué ci-dessus.
Tout comme votre mot de passe classique, ce mot de passe spécifique à une application permet d'accorder un accès complet à votre compte Google. Étant donné que vous n'avez pas besoin de le mémoriser, ne le notez nulle part ni ne le partagez avec personne.

OK

5- Teste du serveur SMTP:

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

```
GNU nano 6.2                               /etc/postfix/sasl/sasl_passwd
[smtp.gmail.com]:587 innokidsenfants@gmail.com:illr moyx ucis kxeh
```

Cette ligne indique à Postfix d'utiliser le serveur SMTP de Gmail sur le port 587 (TLS) pour l'envoi de courrier électronique sortant. Les informations d'identification de l'expéditeur sont fournies sous la forme `adresse_email:mot_de_passe`.

cette ligne dans le fichier `sasl_passwd` indique à Postfix comment se connecter au serveur SMTP de Gmail en utilisant les informations d'identification fournies, ce qui lui permet d'envoyer des e-mails via ce serveur de manière sécurisée et authentifiée.

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

Cette commande indique à Postfix d'utiliser `postmap` pour générer une base de données à partir du fichier `sasl_passwd` situé dans le répertoire `/etc/postfix/sasl/`. Cela crée le fichier `sasl_passwd.db`, qui contient les informations d'identification nécessaires pour l'envoi d'e-mails via le serveur SMTP de Gmail.

Cette étape est importante car Postfix peut accéder plus rapidement aux informations d'identification stockées dans le fichier de base de données lors de l'envoi de courrier électronique. De plus, cela permet une gestion plus efficace des informations d'identification et offre une meilleure sécurité.

```
firdaous@innokids-postfix:~$ sudo nano /etc/postfix/sasl/sasl_passwd
firdaous@innokids-postfix:~$ ls /etc/postfix/sasl
sasl_passwd
firdaous@innokids-postfix:~$ sudo postmap /etc/postfix/sasl/sasl_passwd
firdaous@innokids-postfix:~$ ls /etc/postfix/sasl
sasl_passwd  sasl_passwd.db
firdaous@innokids-postfix:~$
```

Pour renforcer la sécurité de notre configuration Postfix :

```
$ sudo chown root:root /etc/postfix/sasl/sasl_passwd /etc/postfix/sasl/sasl_passwd.db
```

La commande change le propriétaire et le groupe des fichiers sasl_passwd et sasl_passwd.db pour les définir sur root:root, limitant ainsi l'accès à ces fichiers aux seuls utilisateurs ayant des priviléges d'administration (root), renforçant ainsi la sécurité.

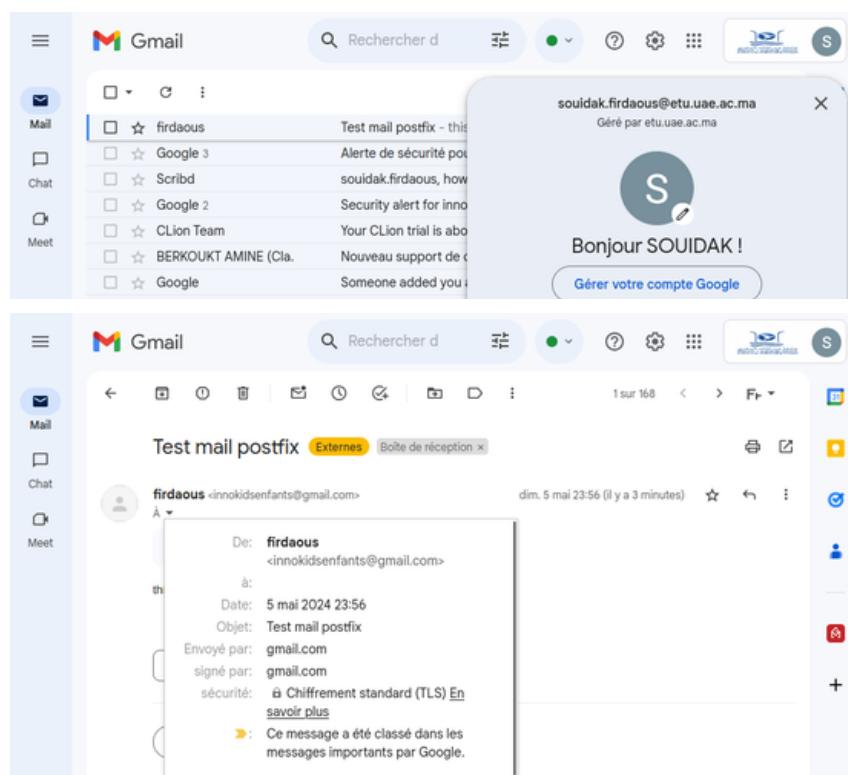
```
$ sudo chmod 0600 /etc/postfix/sasl/sasl_passwd /etc/postfix/sasl/sasl_passwd.db
```

La commande modifie les autorisations des fichiers sasl_passwd et sasl_passwd.db pour les définir sur 0600, permettant uniquement au propriétaire (root) de lire et de modifier ces fichiers. Cela garantit que les informations sensibles, comme les mots de passe, sont protégées contre l'accès non autorisé.

Tester le serveur SMTP:

```
firdaous@innokids-postfix:~$ sendmail soudak.firdaous@etu.uae.ac.ma
sendmail: warning: /etc/postfix/main.cf, line 54: overriding earlier entry: smtp_tls_security_level=m
ay
postdrop: warning: /etc/postfix/main.cf, line 54: overriding earlier entry: smtp_tls_security_level=m
ay
Subject: Test mail postfix
this is just a test email
.

firdaous@innokids-postfix:~$
```



NGINX vs Apache

1- Qu'est-ce qu' Un serveur web ?

Un **serveur web** est un logiciel informatique qui sert du contenu web. Il crée également une connexion entre un serveur informatique et le navigateur web d'un utilisateur, en transmettant des fichiers dans les deux sens.

Le choix du bon serveur web est crucial lors de la création d'un site web ou du lancement d'un projet, car il peut affecter de manière significative les performances et la sécurité d'un site.

Si vous ne savez pas quel serveur web utiliser, pensez à Apache et NGINX. Ces deux serveurs web sont les plus populaires et sont responsables de plus de la moitié du trafic sur Internet.

Donc nous allons vous présenter une comparaison détaillée de NGINX et d'Apache.



2-NGINX vs Apache – Aperçu général

Apache HTTP Server (Apache) :

Apache est un logiciel de serveur web open source, connu pour sa flexibilité et sa compatibilité avec divers systèmes d'exploitation. Il permet de servir des pages web et des applications depuis sa création en 1995.

NGINX :

NGINX est un serveur web open source développé pour répondre aux besoins de gestion efficace de multiples connexions simultanées. Lancé en 2002, il utilise une architecture asynchrone et événementielle, optimisée pour la performance dans la distribution de fichiers statiques et le support d'équilibre de charge.

3- Nginx vs Apache

A-Architecture de base – Gestion des connexions:

L'architecture d'un serveur web est la disposition logique ou le mécanisme qui détermine comment un serveur web traite les demandes, les connexions et le trafic web.

Apache :

Apache suit par défaut une architecture orientée processus, ce qui signifie qu'il crée un seul thread pour traiter chaque demande de connexion.

L'inconvénient de l'architecture orientée processus est qu'Apache doit créer de nombreux processus lorsqu'il traite un grand nombre de demandes. Cela peut entraîner une forte consommation de ressources, provoquant des problèmes de serveur tels qu'un chargement lent des pages Web et des pannes de site.

NGINX :

NGINX adopte une approche différente en utilisant une architecture asynchrone et non bloquante basée sur les événements. Cela permet au serveur web de gérer plusieurs connexions dans un seul processus.

NGINX possède un processus maître qui effectue des opérations privilégiées telles que la liaison aux ports, la lecture et l'évaluation des fichiers de configuration, et la création de plusieurs processus enfants.

B- Comparaison des performances

Les performances d'un serveur web sont généralement déterminées par sa capacité à gérer du contenu statique et dynamique.

Apache :

Apache sert du contenu statique en utilisant son approche traditionnelle basée sur les fichiers.

NGINX :

Lorsqu'il s'agit de servir du contenu statique, NGINX est plus rapide qu'Apache car il met en cache les fichiers statiques pour les rendre disponibles dès qu'ils sont demandés.

C-Sécurité avec Apache et NGINX:

Il est essentiel de choisir un serveur web sûr et fiable, capable de protéger les données de votre site web et régulièrement mis à jour avec les derniers correctifs.

Apache :

La Fondation Apache Software améliore la sécurité du serveur HTTP Apache avec des paramètres pour gérer les attaques DoS, tels que TimeOut, RequestReadTimeout et KeepAliveTimeout. La sécurité dépend aussi de sa configuration, incluant l'utilisation de pare-feu d'application web .

NGINX :

NGINX utilise des contrôles de sécurité comme la limitation du taux pour se protéger contre les attaques DDoS et la gestion de l'accès par IP. Il supporte TLS pour le cryptage des données. NGINX Plus offre des fonctionnalités avancées, comme l'authentification unique (SSO). Les mises à jour de sécurité sont disponibles sur le site de NGINX.

D- Choisir un serveur Web

Après avoir parcouru les comparaisons entre NGINX et Apache, il est clair que chaque logiciel présente des avantages et des inconvénients. Il est donc essentiel de connaître ses besoins avant de choisir entre NGINX et Apache.

Apache est adapté aux environnements d'hébergement mutualisé. Il offre un accès root pour modifier le fichier de configuration principal, ce qui permet aux utilisateurs non privilégiés de contrôler plusieurs aspects du serveur.

L'inconvénient est que ce logiciel peut consommer beaucoup de mémoire du serveur.

Quant à NGINX, il est plus performant qu'Apache pour le traitement des requêtes de contenu statique. Il peut également servir de nombreux clients en même temps lors d'une charge élevée, ce qui en fait un excellent choix pour un site à fort trafic.

Toutefois, ce logiciel ne peut pas servir de contenu dynamique par défaut, et il doit acheminer toutes les demandes de contenu dynamique vers un serveur d'applications dorsal.

3-Apache et NGINX peuvent-ils fonctionner ensemble?

Il est possible de faire fonctionner NGINX et Apache ensemble et de profiter des forces de chaque serveur – NGINX pour sa vitesse de traitement et Apache pour ses modules puissants.

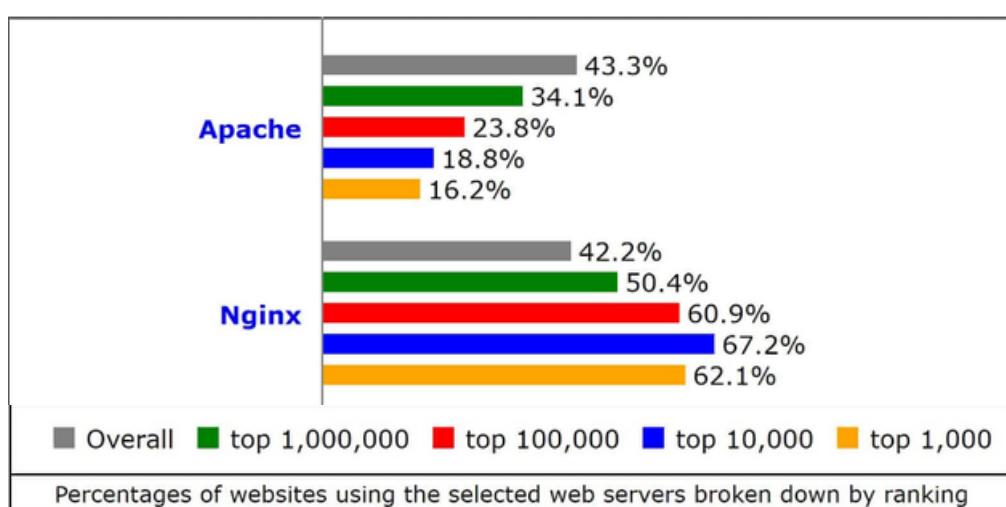
L'utilisation d'Apache et de NGINX peut réduire certains blocages qui se produisent généralement lorsqu'un thread ou un processus Apache est occupé, ce qui contribue à améliorer les performances de votre serveur.

4-Conclusion:

Chaque logiciel a ses avantages et ses inconvénients, de sorte que la décision d'utiliser NGINX ou Apache dépendra entièrement des préférences de l'utilisateur.

Au lieu de choisir soit NGINX soit Apache, il peut être plus efficace d'utiliser les deux logiciels pour améliorer les performances de votre serveur – NGINX comme serveur proxy inverse pour traiter les demandes de contenu statique et Apache comme back-end pour servir le contenu dynamique.

Ces données concernent l'ensemble des serveurs dans le monde:



Nginx

1-Introduction

Dans notre projet, nous avons opté pour NGINX comme serveur web en raison de ses performances élevées et de son efficacité dans la gestion des connexions simultanées. .



Bien que nous ayons effectué une comparaison approfondie entre NGINX et Apache, notre choix s'est porté sur NGINX en raison de ses avantages spécifiques pour notre environnement et nos besoins

2-Installation de Nginx

A- Mettre à Jour les Paquets du Système :

```
$ sudo apt update
```

B- Installation de Nginx :

```
$ sudo apt install nginx
```

Pour connaître la version installée de Nginx on utilise la commande :

```
firdaous@enfants-innova-ma:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
firdaous@enfants-innova-ma:~$
```

pour tester la configuration de Nginx sans redémarrer le serveur

```
firdaous@enfants-innova-ma:~$ sudo nginx -t
[sudo] password for firdaous:
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

cette commande vérifie la syntaxe des fichiers de configuration Nginx pour détecter toute erreur ou incohérence. Si la configuration est correcte, elle affiche un message de confirmation, sinon, elle signale les erreurs trouvées.

C- Les processus gèrent et prennent en charge les nouvelles configurations :

```
$ sudo systemctl status nginx
```

Cette commande permet de vérifier l'état actuel du service Nginx. Elle affiche des informations sur son état, comme s'il est en cours d'exécution, s'il a rencontré des erreurs lors du démarrage, etc.

```
$ sudo systemctl start nginx
```

Cette commande démarre le service Nginx sur notre système. L'utilisation de sudo est nécessaire pour exécuter cette commande avec des privilèges élevés, car elle implique le démarrage d'un service système.

```
firdaous@enfants-innova-ma: ~$ sudo systemctl start nginx
firdaous@enfants-innova-ma: ~$ 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 Wed 2024-05-08 22:59:32 +01; 1s ago
    Docs: man:nginx(8)
 Process: 15893 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited,>
 Process: 15894 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0)
 Main PID: 15895 (nginx)
   Tasks: 13 (limit: 28308)
  Memory: 10.7M
    CPU: 49ms
   CGroup: /system.slice/nginx.service
           ├─15895 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           ├─15896 "nginx: worker process" " "
           ├─15897 "nginx: worker process" " "
           ├─15898 "nginx: worker process" " "
           ├─15899 "nginx: worker process" " "
           ├─15900 "nginx: worker process" " "
           └─15901 "nginx: worker process" " "


```

```
$ sudo systemctl stop nginx
```

Cette commande arrête le service Nginx en cours d'exécution sur votre système. L'utilisation de sudo est nécessaire pour exécuter cette commande avec des privilèges élevés, car elle implique l'arrêt d'un service système.

```
firdaous@enfants-innova-ma: ~$ sudo systemctl stop nginx
[sudo] password for firdaous:
firdaous@enfants-innova-ma: ~$ 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 Wed 2024-05-08 22:57:39 +01; 4s ago
    Docs: man:nginx(8)
 Process: 13760 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited,>
 Process: 13761 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0)
 Process: 15820 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/ngin...
 Main PID: 13862 (code=exited, status=0/SUCCESS)
   CPU: 137ms

22:12:09 08 ↵ enfants.innova.ma systemd[1]: Starting A high performance web server and a reverse p...
22:12:09 08 ↵ enfants.innova.ma systemd[1]: Started A high performance web server and a reverse pr...
22:57:39 08 ↵ enfants.innova.ma systemd[1]: Stopping A high performance web server and a reverse p...
22:57:39 08 ↵ enfants.innova.ma systemd[1]: nginx.service: Deactivated successfully.
22:57:39 08 ↵ enfants.innova.ma systemd[1]: Stopped A high performance web server and a reverse pr...


```

```
$ sudo service nginx reload
```

Cette commande est utilisée pour recharger la configuration de Nginx sans arrêter le service. Nous pouvons utiliser cette commande pour appliquer les modifications sans interrompre le service en cours d'exécution.

```
$ sudo service nginx restart
```

Cette commande est utilisée pour redémarrer le service Nginx. Lorsque nous exécutons cette commande, Nginx s'arrête d'abord, puis redémarre avec la nouvelle configuration

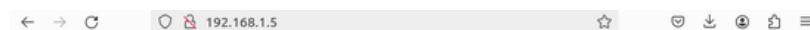
D- pages d'accueil par défaut de NGINX

Affichage des Interfaces Réseau utilisées sur notre machine avec la commande:

```
$ ip a
```

```
Firdaous@enfants-lnova-m:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s31f6: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether e0:73:e7:c0:bc:03 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.5/24 brd 192.168.1.255 scope global enp0s31f6
            valid_lft forever preferred_lft forever
3: wlp0s20f3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether ac:19:8e:9b:ad:75 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.102/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp0s20f3
            valid_lft 258976sec preferred_lft 258976sec
        inet6 fe80::5e13:ae87:aabc:4643/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

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**. réussi sur deux adresses IP locales, 192.168.1.5 et 192.168.1.102, ce qui confirme que le serveur est correctement installé .

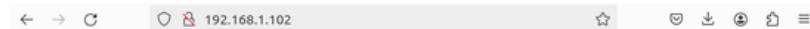


Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

3-Configuration de Nginx

A-Configuration du Pare-feu UFW

Pourquoi faisons-nous la Configuration du Pare-feu UFW ?

Nous activons et configurons le pare-feu UFW pour renforcer la sécurité de notre système en limitant le trafic entrant uniquement aux ports et protocoles nécessaires. Autoriser spécifiquement le trafic sur le port 80 permet de rendre notre serveur Web accessible depuis Internet, tout en restreignant l'accès aux autres ports et services. Cela réduit la surface d'attaque potentielle de notre système, protégeant ainsi notre serveur contre les tentatives d'accès non autorisées ou les attaques malveillantes.

Pour activer UFW et appliquer des règles de pare-feu, suivez ces étapes :

Activer le pare-feu UFW :

```
$ sudo ufw enable
```

Autoriser le trafic sur le port 80 :

```
$ sudo ufw allow 80/tcp
```

Vérifier l'état du pare-feu :

```
$ sudo ufw status
```

```
firdaous@enfants-innova-ma:~$ sudo ufw status
Status: inactive
firdaous@enfants-innova-ma:~$ sudo ufw enable
Firewall is active and enabled on system startup
firdaous@enfants-innova-ma:~$ sudo ufw status
Status: active
firdaous@enfants-innova-ma:~$ sudo ufw allow 80/tcp
Rule added
Rule added (v6)
firdaous@enfants-innova-ma:~$ sudo ufw status
Status: active

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

B- Les fichiers de configuration

Les fichiers de configuration de Nginx se trouvent dans le répertoire suivant :

```
$ cd /etc/nginx/
```

```
firdaous@enfants-innova-ma:~$ cd /etc/nginx
firdaous@enfants-innova-ma:/etc/nginx$ ls
conf.d      koi-utf    modules-available  proxy_params   sites-enabled  win-utf
fastcgi.conf  koi-win    modules-enabled   scgi_params   snippets
fastcgi_params mime.types nginx.conf     sites-available uwsgi_params
firdaous@enfants-innova-ma:/etc/nginx$
```

Ce répertoire contient les fichiers de configuration suivant :

- **nginx.conf** : Fichier principal de configuration de NGINX, contenant les paramètres globaux et les directives de fonctionnement du serveur.
- **sites-available** et **sites-enabled** : Répertoires utilisés pour gérer les configurations des sites web. sites-available contient toutes les configurations possibles, tandis que sites-enabled contient des liens symboliques vers les configurations activées, facilitant ainsi la gestion des multiples sites sur un serveur.
- **conf.d** : Répertoire pour les fichiers de configuration additionnels pouvant être inclus dans la configuration principale de NGINX, permettant d'ajouter des configurations spécifiques à des applications ou modules.
- **mime.types** : Fichier définissant les types MIME associés aux extensions de fichier, crucial pour la gestion des contenus envoyés par NGINX aux navigateurs.
- **modules-available** et **modules-enabled** : Ces répertoires gèrent les modules disponibles et activés dans NGINX, facilitant l'extension des fonctionnalités du serveur selon les besoins.
- **snippets** : Répertoire contenant des fragments de configuration réutilisables, permettant d'éviter la duplication de code et facilitant la gestion des configurations communes.

Ces éléments structurent la configuration de NGINX de manière modulaire et organisée, rendant plus simple la gestion, la personnalisation et la maintenance d'un serveur web NGINX adapté aux besoins spécifiques de votre application ou environnement.

C- Configuration du serveur:

Accéder au répertoire sites-available et Créer un fichier de configuration pour le site:

```
$ cd /etc/nginx/sites-available/  
$ sudo nano enfants.conf
```

```
firdaous@enfants-innova-ma:/etc/nginx$ cd sites-available/  
firdaous@enfants-innova-ma:/etc/nginx/sites-available$ ls  
default  
firdaous@enfants-innova-ma:/etc/nginx/sites-available$ sudo nano eanfants.conf  
[sudo] password for firdaous:  
firdaous@enfants-innova-ma:/etc/nginx/sites-available$
```

Accéder au répertoire sites-enabled :

```
$ cd /etc/nginx/sites-enabled/
```

Le répertoire sites-enabled contient des liens symboliques vers les fichiers de configuration des sites Web activés dans Nginx. Dans notre cas, il y a un seul lien symbolique nommé default, qui pointe vers le fichier de configuration du site par défaut dans sites-available. Cela signifie que le site par défaut est activé sur le serveur Nginx.

```
firdaous@enfants-innova-ma:/etc/nginx/sites-available$ cd ..  
firdaous@enfants-innova-ma:/etc/nginx$ cd sites-enabled/  
firdaous@enfants-innova-ma:/etc/nginx/sites-enabled$ ls -l  
total 0  
lrwxrwxrwx 1 root root 34 22:12 8      ./.default -> /etc/nginx/sites-available/default  
firdaous@enfants-innova-ma:/etc/nginx/sites-enabled$
```

créer un lien symbolique vers le fichier de configuration enfants.conf du répertoire sites-available dans le répertoire sites-enabled.

```
firdaous@enfants-innova-ma:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/enfants.conf  
./.enfants.conf  
firdaous@enfants-innova-ma:/etc/nginx/sites-enabled$ ls -l  
total 0  
lrwxrwxrwx 1 root root 34 22:12 8      ./.default -> /etc/nginx/sites-available/default  
lrwxrwxrwx 1 root root 39 01:14 12      ./.enfants.conf -> /etc/nginx/sites-available/enfants.conf  
firdaous@enfants-innova-ma:/etc/nginx/sites-enabled$
```

Le fichier principal de configuration de NGINX est nginx.conf :

```
$ cd /etc/nginx/  
$ sudo nano nginx.conf
```

```

GNU nano 6.2                               nginx.conf
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;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    ##
    # SSL Settings
    ##

    ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
    ssl_prefer_server_ciphers on;

    ##
    # Logging Settings
    ##

    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;

    ##
    # Gzip Settings
    ##

    gzip on;

    # gzip_vary on;
    # gzip_proxied any;
    # gzip_comp_level 6;
    # gzip_buffers 16 8k;
    # gzip_http_version 1.1;
    # gzip_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

    ##
    # Virtual Host Configs
    ##

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

```

Pour inclure le fichier de configuration enfants.conf dans notre configuration principale Nginx (nginx.conf), Nous devons modifié la directive include appropriée dans la section http de notre fichier nginx.conf.
le fichier de configuration enfants.conf est situé dans le répertoire /etc/nginx/sites-enabled/ .

```

## 
# Virtual Host Configs
## 

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

D-Partie http (Port 80):

Accéder au fichier de configuration pour le site enfants.conf:

```
$ cd /etc/nginx/sites-available/
$ sudo nano enfants.conf
```

```

GNU nano 6.2                                     enfants.conf
server {
    listen 80;

    server_name www.enfants.innova.ma enfants.innova.ma;

    root /var/www/enfants;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}

```

Ce fichier de configuration Nginx met en place un serveur web avec les caractéristiques suivantes :

- **Port d'écoute** : Le serveur écoute les requêtes **HTTP** sur **le port 80**, qui est le port par défaut pour les connexions HTTP non sécurisées.
- **Noms de domaine** : Le serveur est configuré pour répondre aux requêtes adressées aux domaines **www.enfants.innova.ma** et **enfants.innova.ma**.
- **Répertoire racine** : Le répertoire où sont stockés les fichiers à servir est **/var/www/enfants**. C'est à partir de ce répertoire que le serveur web récupère les fichiers demandés par les utilisateurs.
- **Fichier d'index** : Le serveur utilise le fichier **index.html** comme page d'accueil par défaut. Cela signifie que lorsque l'utilisateur accède à un répertoire sans spécifier de fichier particulier, le serveur renvoie le fichier **index.html** de ce répertoire.
- **Gestion des requêtes** : Le serveur tente de trouver un fichier ou un répertoire correspondant à l'URI (Uniform Resource Identifier) demandé par l'utilisateur. Si le fichier ou le répertoire n'existe pas, le serveur renvoie **une erreur 404** (fichier non trouvé).

Contenu:

Accéder au `/var/www` et créer un dossier `enfants` et un fichier `index.html`

```

$ cd /var/www
$ sudo mkdir enfants
$ sudo nano index.html

```

```

firdaous@enfants-innova-ma:/etc/nginx/sites-enabled$ cd /var/www/
firdaous@enfants-innova-ma:/var/www$ sudo mkdir enfants
firdaous@enfants-innova-ma:/var/www$ ls -l
total 8
drwxr-xr-x 2 root root 4096 01:20 12    مجلہ enfants
drwxr-xr-x 2 root root 4096 20:38 11    مجلہ html
firdaous@enfants-innova-ma:/var/www$ cd enfants
firdaous@enfants-innova-ma:/var/www/enfants$ sudo nano index.html
firdaous@enfants-innova-ma:/var/www/enfants$ 

```

```

GNU nano 6.2                               index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Page de test NGINX</title>
</head>
<body>
    <h1>Bienvenue sur votre page de test NGINX !</h1>
    <p>Cette page est destinée à tester le fonctionnement de votre serveur NGINX.</p>
    <p>Si vous voyez ce message, cela signifie que NGINX fonctionne correctement et sert cette page web.</p>
    <p>N'hésitez pas à modifier cette page pour tester d'autres fonctionnalités de NGINX ou à y inclure du contenu personnalisé.</p>
</body>
</html>

```

Nous avons redémarré NGINX:

```

firdaous@enfants-innova-ma:~/var/www/enfants$ sudo systemctl restart nginx
firdaous@enfants-innova-ma:~/var/www/enfants$ 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 2024-05-12 01:29:19 +01; 6s ago
       Docs: man:nginx(8)
   Process: 25257 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, st>
   Process: 25258 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU>
 Main PID: 25259 (nginx)
   Tasks: 13 (limit: 28308)
     Memory: 10.8M
        CPU: 45ms
      CGroup: /system.slice/nginx.service
              ├─25259 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
              ├─25260 "nginx: worker process" └─25261 "nginx: worker process" └─25262 "nginx: worker process" └─25263 "nginx: worker process" └─25264 "nginx: worker process" └─25265 "nginx: worker process" └─25266 "nginx: worker process" └─25267 "nginx: worker process" └─25268 "nginx: worker process" └─25269 "nginx: worker process" └─25270 "nginx: worker process" └─25271 "nginx: worker process"
              01:29:19.12 12.d.enfants.innova.ma systemd[1]: Starting A high performance web server and a reverse proxy...
              01:29:19.12.d.enfants.innova.ma systemd[1]: Started A high performance web server and a reverse proxy...
Lines 1-27/27 (END)

```

Maintenant, si nous accédons à notre site à l'aide de l'URL que nous avons configurée (<http://enfants.innova.ma:80/>), nous devrions voir notre page index.html nouvellement modifiée.



La résolution DNS de www.enfants.innova.ma pointe vers l'adresse IP 192.229.221.95. Les tests de ping vers cette adresse montrent une communication stable avec le serveur. Cela confirme que le serveur NGINX répond correctement aux requêtes adressées à www.enfants.innova.ma.

```

firdaous@enfants-innova-ma:~$ ping www.enfants.innova.ma
PING www.enfants.innova.ma (192.229.221.95) 56(84) bytes of data.
64 bytes from www.enfants.innova.ma (192.229.221.95): icmp_seq=1 ttl=54 time=51.3 ms
64 bytes from www.enfants.innova.ma (192.229.221.95): icmp_seq=2 ttl=54 time=50.9 ms
64 bytes from www.enfants.innova.ma (192.229.221.95): icmp_seq=3 ttl=54 time=50.9 ms
64 bytes from www.enfants.innova.ma (192.229.221.95): icmp_seq=4 ttl=54 time=51.3 ms
64 bytes from www.enfants.innova.ma (192.229.221.95): icmp_seq=5 ttl=54 time=50.8 ms
^C
--- www.enfants.innova.ma ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 50.756/51.018/51.307/0.220 ms
firdaous@enfants-innova-ma:~$ 

```

E- Partie https (Port 443):

Accéder au fichier de configuration pour le site enfants.conf:

```
$ cd /etc/nginx/sites-available/  
$ sudo nano enfants.conf
```

```
GNU nano 6.2                                         enfants.conf  
server {  
    listen 80;  
    listen [::]:80;  
    listen 443 ssl;  
    listen [::]:443 ssl;  
  
    ssl_certificate /etc/nginx/ssl-certs/enfants.crt;  
    ssl_certificate_key /etc/nginx/ssl-certs/enfants.key;  
  
    server_name www.enfants.innova.ma enfants.innova.ma;  
  
    root /var/www/enfants;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Les modifications apportées au configuration Nginx par rapport au premier sont les suivantes :

- **Ajout de l'écoute sur le port 443 pour SSL/TLS :**
 - Ajoute l'écoute sur le port 443, qui est le port standard pour les connexions HTTPS sécurisées. Cela permet au serveur de gérer les connexions sécurisées en plus des connexions HTTP non sécurisées.
- **Support de l'IPv6 :**
 - L4inclut des directives pour écouter sur les adresses IPv6 ([::]:80 et [::]:443). Cela permet au serveur de gérer les connexions IPv6 sur les ports 80 et 443.
- **Configuration SSL :**
 - Le bloc configure SSL/TLS pour sécuriser les connexions. Il spécifie l'emplacement du certificat SSL (ssl_certificate /etc/nginx/ssl-certs/enfants.crt) et du certificat de confiance (ssl_trusted_certificate /etc/nginx/ssl-certs/enfants.crt). Ces directives sont nécessaires pour établir des connexions HTTPS sécurisées.

Générer un certificat SSL:

```
firdaous@enfants-innova-ma:/etc/nginx/sites-available$ cd ..  
firdaous@enfants-innova-ma:/etc/nginx$ sudo mkdir ssl-certs  
firdaous@enfants-innova-ma:/etc/nginx$ cd ssl-certs  
firdaous@enfants-innova-ma:/etc/nginx/ssl-certs$ ls -l  
total 0  
firdaous@enfants-innova-ma:/etc/nginx/ssl-certs$
```

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

Cette commande générera un certificat auto-signé valide pour 365 jours et le sauvegardera dans les fichiers enfants.crt et enfants.key dans le répertoire /etc/nginx/ssl-certs/. Pendant la génération du certificat, nous serons invité à fournir des informations sur notre organisation, le nom commun du certificat, etc.

F- Utilisation de la commande ss -ltn pour lister les connexions TCP en écoute:

\$ ss -ltn

Ces résultats montrent les sockets TCP en écoute sur votre système :

- **0.0.0.0:80** : Cette entrée signifie qu'il y a une écoute sur toutes les interfaces (0.0.0.0) sur le port 80. Cela indique généralement un serveur web qui écoute sur le port HTTP standard.
- **0.0.0.0:25** : Cette entrée indique une écoute sur toutes les interfaces sur le port 25. Le port 25 est généralement utilisé pour le service de courrier SMTP (Simple Mail Transfer Protocol).
- **0.0.0.0:443** : Cette entrée indique une écoute sur toutes les interfaces sur le port 443. Le port 443 est généralement utilisé pour les connexions HTTPS sécurisées, ce qui suggère un serveur web sécurisé (HTTPS).

Ces informations indiquent que votre système écoute sur les ports 80, 25 et 443. Cela peut correspondre à un serveur web (HTTP et HTTPS) ainsi qu'à un serveur de messagerie SMTP.

G- Autorisation du Trafic HTTPS sur le Port 443 avec UFW:

```
$ sudo ufw allow 443/tcp
```

```
firdaous@enfants-innova-ma:/etc/nginx/ssl-certs$ sudo ufw status
Status: active

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

firdaous@enfants-innova-ma:/etc/nginx/ssl-certs$ sudo ufw allow 443/tcp
Rule added
Rule added (v6)
firdaous@enfants-innova-ma:/etc/nginx/ssl-certs$
```

4- Test Initial du Site Web



Warning: Potential Security Risk Ahead

Firefox detected a potential security threat and did not continue to **enfants.innova.ma**. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

[Learn more...](#)

[Go Back \(Recommended\)](#)

[Advanced...](#)

enfants.innova.ma uses an invalid security certificate.

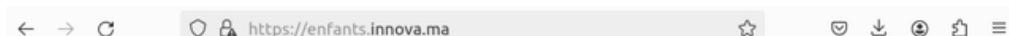
The certificate is not trusted because it is self-signed.

Error code: [MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT](#)

[View Certificate](#)

[Go Back \(Recommended\)](#)

[Accept the Risk and Continue](#)

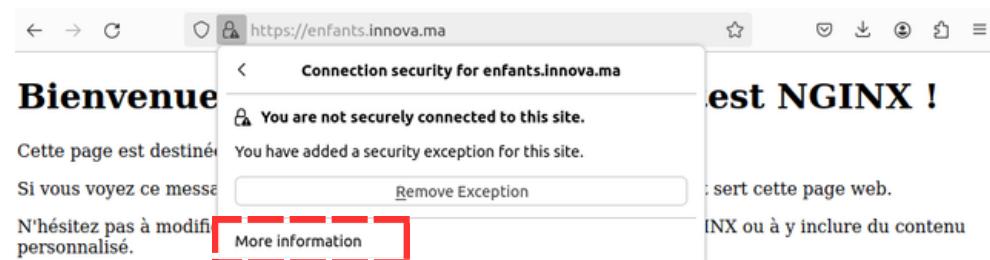


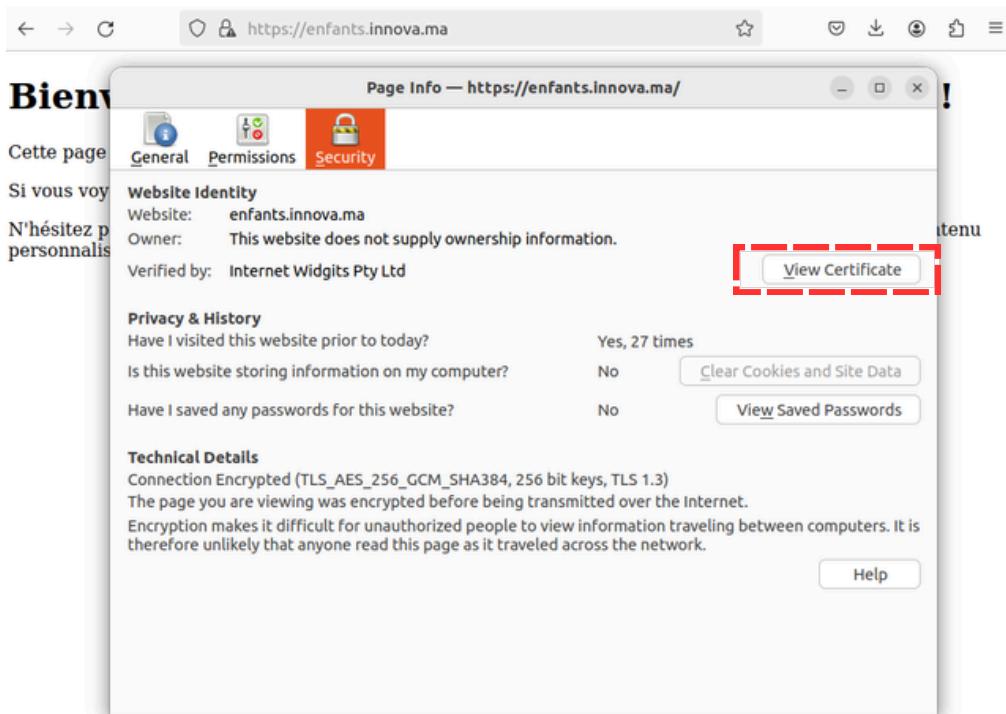
Bienvenue fir fir sur votre page de test NGINX !

Cette page est destinée à tester le fonctionnement de votre serveur NGINX.

Si vous voyez ce message, cela signifie que NGINX fonctionne correctement et sert cette page web.

N'hésitez pas à modifier cette page pour tester d'autres fonctionnalités de NGINX ou à y inclure du contenu personnalisé.





obtenir et installer le certificat SSL, permettant ainsi des connexions sécurisées via HTTPS pour votre site web.

Certificate

The screenshot shows a certificate viewer for the domain `enfants.innova.ma`. The certificate details are as follows:

- Subject Name:**

Country	MR
State/Province	Some-State
Organization	Internet Widgits Pty Ltd
Common Name	enfants.innova.ma
Email Address	innokidsenfants@gmail.com
- Issuer Name:**

Country	MR
State/Province	Some-State
Organization	Internet Widgits Pty Ltd
Common Name	enfants.innova.ma
Email Address	innokidsenfants@gmail.com
- Validity:**

Not Before	Sun, 12 May 2024 23:52:44 GMT
Not After	Mon, 12 May 2025 23:52:44 GMT
- Public Key Info:**

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	<code>CF:E4:CF:0A:22:8E:3A:C2:88:09:83:C6:A8:0C:87:A0:56:FA:0D:65:6C:8C:94:1C:...</code>

5-Déploiement de l'application

A- Accéder au Répertoire du Projet :

```
firdaous@enfants-innova-ma:~/InnoKids-develop
firdaous@enfants-innova-ma:~/InnoKids-develop$ cd frontend/
firdaous@enfants-innova-ma:~/InnoKids-develop/frontend$ ls
conf           index.html      LICENSE          package-lock.json  public      src    tailwind.config.js
dockerfile     jsconfig.json   package.json    postcss.config.js README.md  ssl    vite.config.js
tsconfig.json
```

B- Installation des Dépendances :

```
$ npm install
```

La commande npm install est utilisée pour installer toutes les dépendances listées dans le fichier package.json.

```
firdaous@enfants-innova-ma:~/InnoKids-develop/frontend$ npm install
up to date, audited 263 packages in 666ms
72 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

C- Construction du Projet :

```
$ npm run build
```

```
firdaous@enfants-innova-ma:~/InnoKids-develop/frontend$ npm run build
> admin-one-vue-tailwind@3.3.0 build
> vite build
vite v4.5.3 building for production...
transforming (33) src/views/Groupe/AjouterGroupe.vueBrowserslist: caniuse-lite is outdated. Please run
npx caniuse-lite <your-file> --update
```

Accès au répertoire de distribution (dist) :

```
firdaous@enfants-innova-ma:~/InnoKids-develop/frontend$ cd dist
firdaous@enfants-innova-ma:~/InnoKids-develop/frontend/dist$ ls
123.jpg  abc.jpg   data.json   ge2.jpg   ge6.jpg   Kids.json   'Project Logo.jpg'
1.json   assets     data-sources  ge3.jpg   image1.jpg  nvm.jpg   robotics.jpg
2.jpg   Chin.jpg   favicon.png  ge4.jpg   index.html offre.png  Teachers
3.jpg   Curriculum  ge1.jpg   ge5.jpg   InnoLogo.jpg  programming.jpg
```

D- Copie du répertoire de distribution :

```
$ sudo cp -r dist /var/www/enfants/
```

E- Vérification du contenu copié :

```
firdaous@enfants-innova-ma:/var/www/enfants$ ls  
dist
```

F- Modification de la configuration Nginx :

```
firdaous@enfants-innova-ma:/var/www/enfants$ cd  
firdaous@enfants-innova-ma:~$ cd /etc/nginx/sites-available/  
firdaous@enfants-innova-ma:/etc/nginx/sites-available$ sudo nano enfants.conf  
firdaous@enfants-innova-ma:/etc/nginx/sites-available$ cd
```

```
GNU nano 6.2                                     enfants.conf  
server {  
    listen 80;  
    listen [::]:80;  
    listen 443 ssl;  
    listen [::]:443 ssl;  
  
    ssl_certificate /etc/nginx/ssl-certs/enfants.crt;  
    ssl_certificate_key /etc/nginx/ssl-certs/enfants.key;  
  
    server_name www.enfants.innova.ma enfants.innova.ma;  
  
    root /var/www/enfants/dist;  
        index index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Racine du site (root /var/www/enfants/dist;) :

Cette ligne indique à Nginx où se trouvent les fichiers du site web. La directive root spécifie que Nginx servira les fichiers depuis le répertoire /var/www/enfants/dist. Par exemple, une requête pour <http://www.enfants.innova.ma/index.html> fera que Nginx cherchera index.html dans ce répertoire. Cette configuration centralise l'emplacement des fichiers du site, facilitant leur gestion.

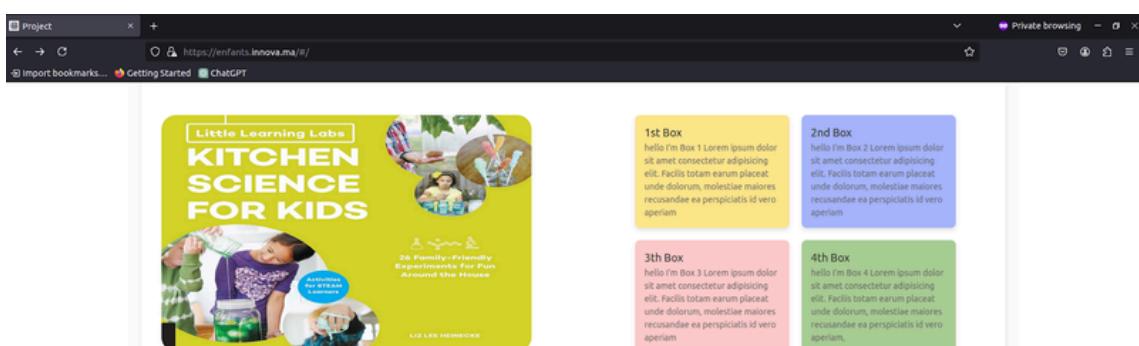
Index par défaut (index index.html;) :

Cette directive spécifie quel fichier servir lorsque Nginx reçoit une requête pour un répertoire sans fichier spécifié. Avec index index.html;, une requête pour <http://www.enfants.innova.ma/> fera que Nginx servira le fichier index.html du répertoire /var/www/enfants/dist, offrant une navigation intuitive aux utilisateurs.

G- Redémarrage de Nginx :

```
firdaous@enfants-innova-ma:~/var/www/enfants$ sudo systemctl restart nginx
firdaous@enfants-innova-ma:~/var/www/enfants$ 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 2024-05-12 01:29:19 +01; 6s ago
      Docs: man:nginx(8)
   Process: 25257 ExecStartPre=/usr/sbin/nginx -t -q daemon on; master_process on; (code=exited, st>
  Process: 25258 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU>
 Main PID: 25259 (nginx)
   Tasks: 13 (limit: 28308)
     Memory: 10.8M
        CPU: 45ms
       CGroup: /system.slice/nginx.service
           ├─25259 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
           ├─25260 "nginx: worker process" └─25261 "nginx: worker process"
           ├─25262 "nginx: worker process" └─25263 "nginx: worker process"
           ├─25264 "nginx: worker process" └─25265 "nginx: worker process"
           ├─25266 "nginx: worker process" └─25267 "nginx: worker process"
           ├─25268 "nginx: worker process" └─25269 "nginx: worker process"
           ├─25270 "nginx: worker process" └─25271 "nginx: worker process"
01:29:19 12.d.e.enfants.innova.ma systemd[1]: Starting A high performance web server and a reverse proxy...
01:29:19 12.d.e.enfants.innova.ma systemd[1]: Started A high performance web server and a reverse proxy...
lines 1-27/27 (END)
```

H- Affichage :



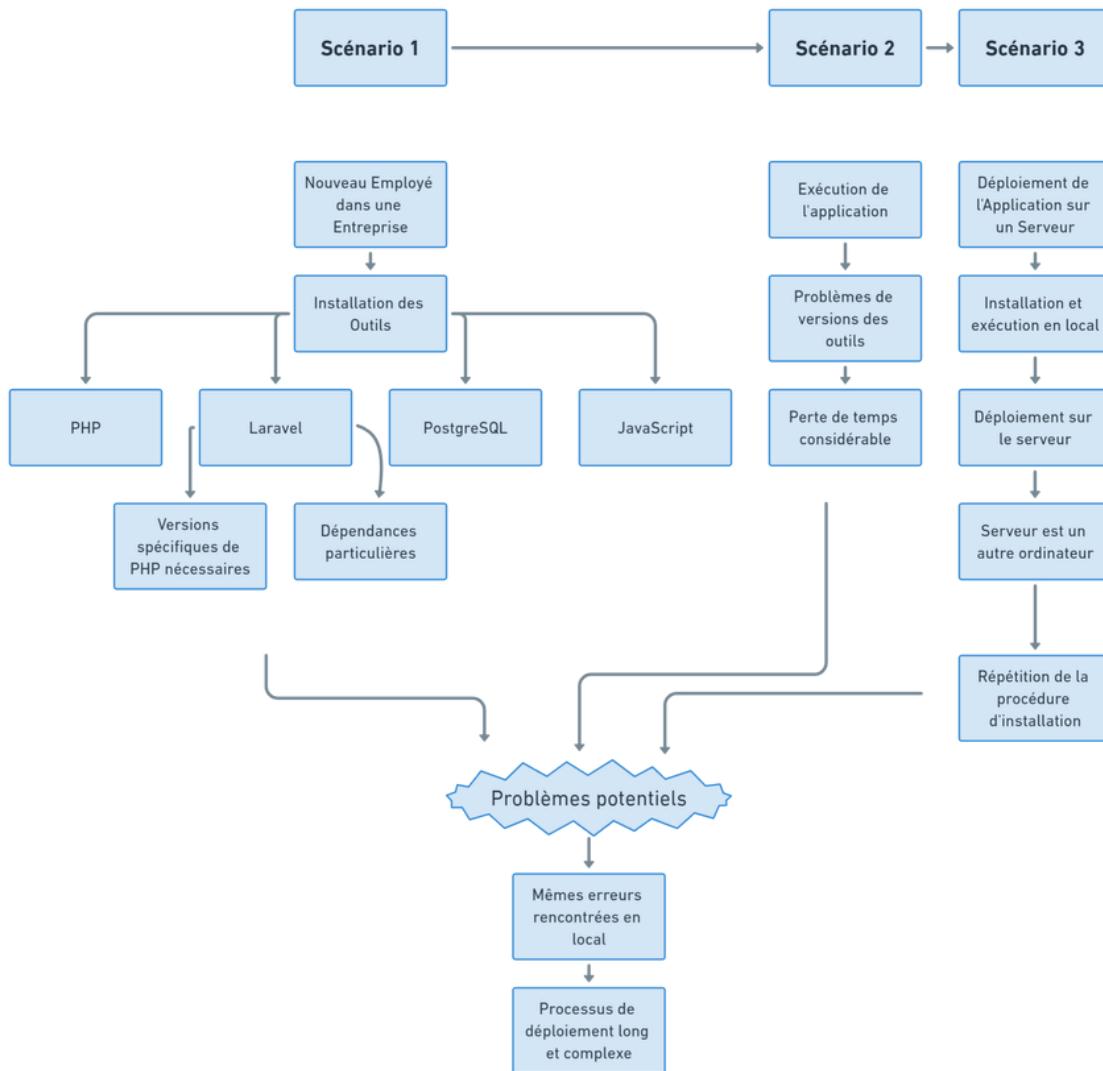
Ateliers de créativité numérique

Des activités extra-scolaires innovantes!



DOCKER

A-Avant Docker?



Problématique

Lorsqu'un nouveau développeur rejoint une entreprise et commence à travailler sur un projet, plusieurs outils doivent être installés, tels que PHP, Laravel, PostgreSQL, JavaScript, etc. Cette installation peut entraîner des erreurs dues aux exigences spécifiques de chaque outil, telles que des versions incompatibles et des dépendances variées. Ces problèmes peuvent causer des pertes de temps significatives. De plus, lors du déploiement de l'application sur un serveur, les mêmes erreurs peuvent survenir, nécessitant une réplication des démarches d'installation sur la machine locale et le serveur.

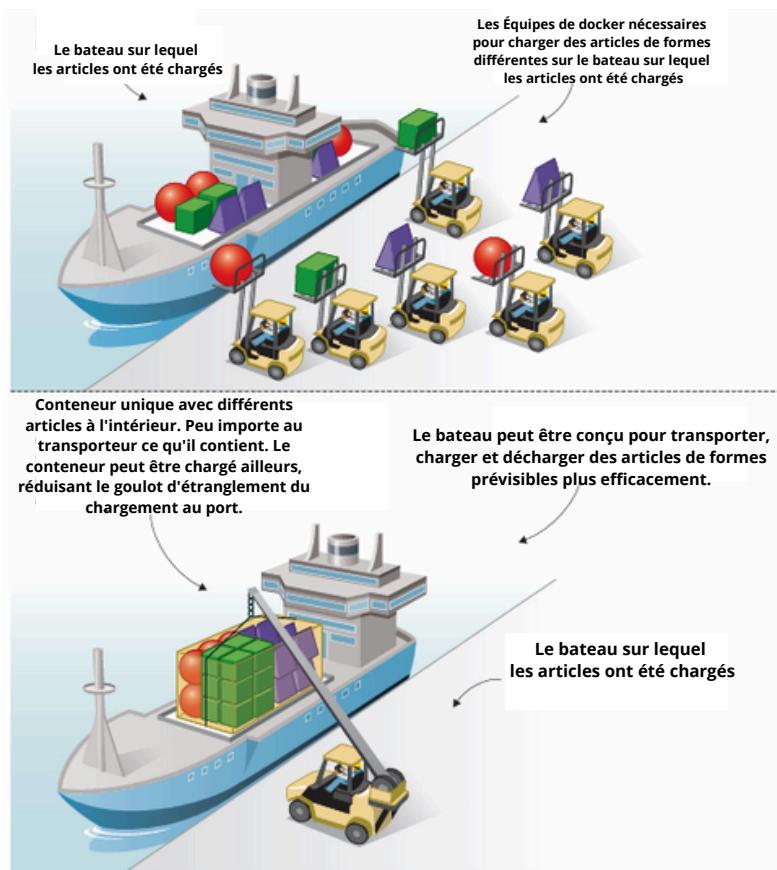


B-Qu'est ce que c'est Docker?

Comme le suggère son logo représentant des conteneurs sur un bateau, Docker facilite le transport d'applications d'un environnement à un autre, à l'instar des marchandises transportées par bateau d'un point X à un point Y. Cette plateforme open source de conteneurisation permet aux développeurs de "charger" leurs applications dans des conteneurs, regroupant ainsi le code et toutes ses dépendances nécessaires. Ces conteneurs standardisés assurent que l'application **fonctionne de manière fiable** et cohérente, dans **n'importe quel environnement**, que ce soit sur le poste de travail d'un développeur ou dans un environnement de production sur un serveur distant. Docker simplifie le déploiement et la gestion des applications en **isolant les applications les unes des autres**, ce qui **minimise les conflits entre systèmes et accélère le lancement de nouvelles versions**.

Cet outil est donc essentiel pour les développeurs et les administrateurs système qui visent une amélioration continue de l'efficacité, de la cohérence et de la scalabilité des applications qu'ils gèrent.

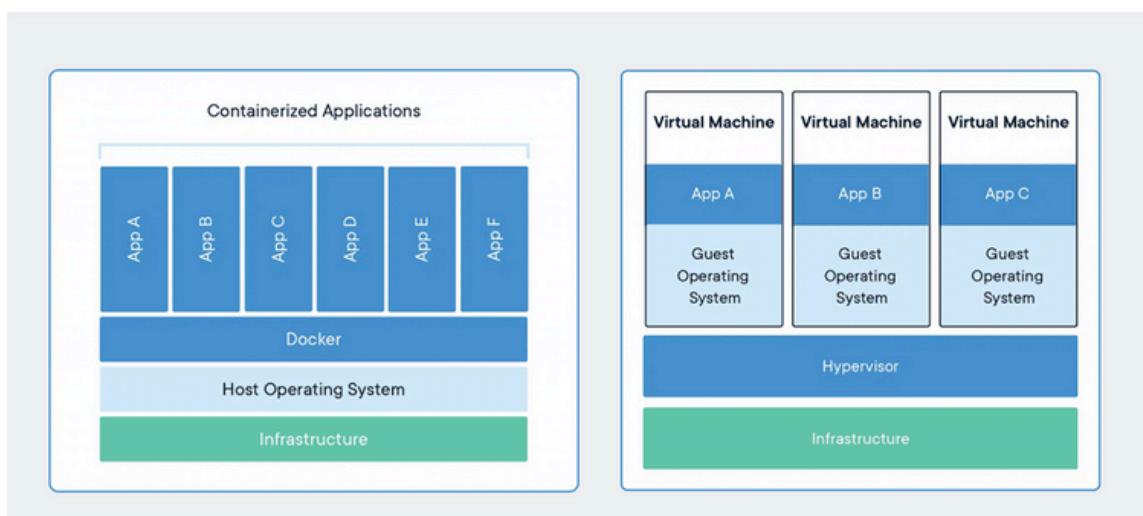
En résumé, Docker offre une solution robuste pour **transporter et exécuter** des applications dans **divers environnements** informatiques **sans craindre les incompatibilités** ou les erreurs de configuration.



C-Docker vs Machines virtuelles

Dans Docker, nous pouvons emballer une application dans une unité standardisée, qui peut être utilisée pour le développement de logiciels. Cette unité, ou conteneur, comprend le codage et les dépendances de l'application afin qu'elle puisse facilement fonctionner sur n'importe quel environnement informatique.

Avant Docker, les entreprises utilisaient souvent des **machines virtuelles** (Virtual Machine ou VM) pour exécuter des applications. Celles-ci peuvent émuler des ordinateurs physiques, permettant aux développeurs de transformer un serveur en plusieurs serveurs. Toutefois, cette approche peut présenter certains **inconvénients**. Chaque VM contient une **copie complète du système d'exploitation et de l'application**, ainsi que les binaires et bibliothèques nécessaires. Ces fichiers peuvent **occuper des dizaines de Go** sur un ordinateur. En outre, la virtualisation du matériel pour un système d'exploitation invité peut **nécessiter des frais généraux importants**. Au lieu de virtualiser le matériel, les conteneurs virtualisent le système d'exploitation. Dans **Docker**, les conteneurs sont des abstractions au niveau de la couche applicative qui peuvent **contenir à la fois du code et des dépendances**. Sur la même machine, plusieurs conteneurs peuvent fonctionner comme des processus isolés :



Par conséquent, les conteneurs Docker prennent généralement **moins d'espace**. Ils peuvent également **contenir plus d'applications sans autant de VM et d'OS**.

D- Comment fonctionne Docker ?

D-1. Image Docker

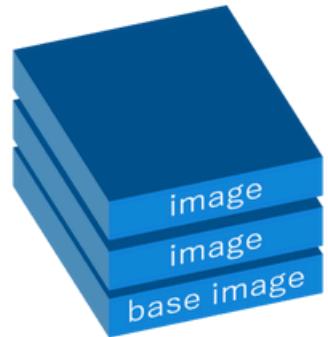
Une image Docker est un paquet logiciel exécutable qui **contient tout le code source, les dépendances et les outils nécessaires pour faire fonctionner une application.** Les images sont des **modèles en lecture seule** qui **fournissent les instructions nécessaires pour créer un conteneur.**

Pour mieux comprendre les images Docker, imaginons les comme des plans. Elles capturent des instantanés de ce que contiendra un conteneur lorsqu'il sera exécuté.

Une image est **composée de plusieurs couches empilées.**

Par exemple, si vous construisez une image de serveur web, vous pouvez commencer par inclure Ubuntu Linux, puis ajouter les couches d'Apache et de PHP.

Lors de la création d'une image Docker, il est **conseillé de placer les couches les plus susceptibles de changer en haut de la pile.** De cette façon, si des modifications sont nécessaires, vous n'aurez pas à reconstruire l'intégralité de l'image, ce qui facilite les mises à jour.



D-2. Conteneur Docker

Comme nous l'avons vu précédemment, les conteneurs sont des éléments essentiels de Docker. Les conteneurs Docker sont des composants clés de Docker qui fonctionnent comme des **zones isolées où une application peut s'exécuter sans impacter le reste du système.** Ces unités logicielles, qui contiennent à la fois le code et toutes les dépendances nécessaires, facilitent le transfert des applications d'un environnement à un autre.

Voici quelques avantages clés de l'utilisation des conteneurs Docker :

- **Standardisation** : Docker a établi une norme industrielle pour l'utilisation des conteneurs, qui existent depuis des décennies. Ces conteneurs sont extrêmement portables et simples à utiliser.
- **Légèreté** : Les conteneurs partagent le noyau du système d'exploitation, évitant la nécessité d'avoir un système d'exploitation distinct pour chaque application. Cela permet une exécution plus efficace des applications tout en réduisant les coûts associés aux serveurs et aux licences.

- **Sécurité** : Contrairement aux machines virtuelles, les conteneurs offrent une isolation qui sécurise les applications en les séparant les unes des autres. Docker assure une isolation par défaut, renforçant la sécurité de vos applications.

D-3.Docker Registry

Le Docker Registry est un **service de stockage et de distribution pour les images Docker**. Il permet aux utilisateurs de pousser et de tirer des images Docker, facilitant le partage et la collaboration sur des projets logiciels. **exemple: Docker Hub** est un registry public maintenu par Docker où vous pouvez **télécharger** des images fournies par d'autres développeurs et organisations, ou **stocker** les vôtres. Les registries privés peuvent également être configurés pour sécuriser les images sensibles et contrôler l'accès. En somme, le Docker Registry est essentiel pour gérer efficacement les images Docker dans un **environnement collaboratif**.

D-4.Docker Client

Le Docker client, souvent simplement appelé "Docker", est **l'interface en ligne de commande** que les développeurs utilisent pour **interagir avec Docker**.

Il permet aux utilisateurs de lancer des commandes pour construire, exécuter, et gérer des conteneurs Docker à partir de leur terminal.

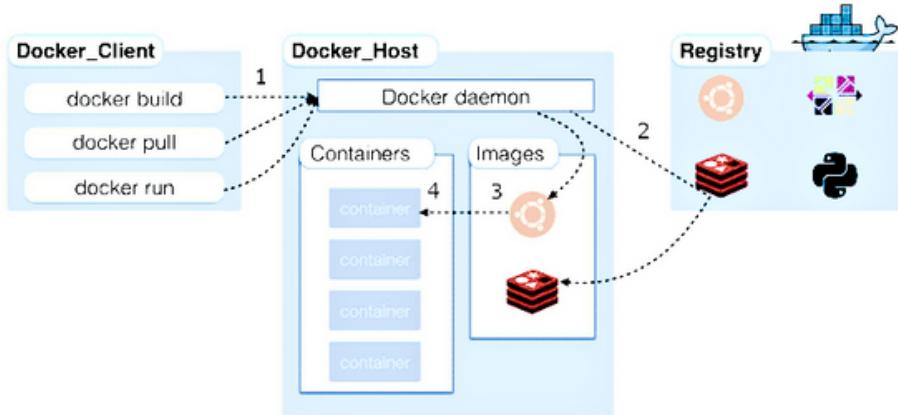
En communiquant avec le Docker Daemon, qui exécute les commandes et gère les conteneurs Docker, le client permet aux développeurs de simplifier et d'automatiser les tâches de déploiement d'applications.

Le Docker client est donc un outil crucial pour manipuler et orchestrer les environnements conteneurisés de manière efficace.

D-5.Docker Daemon

Le Docker Daemon est un processus en arrière-plan qui gère les conteneurs Docker. Il écoute et exécute les commandes envoyées par le Docker Client, réalisant ainsi les opérations nécessaires au fonctionnement des conteneurs. En tant que moteur central de Docker, il joue un rôle crucial dans l'orchestration et l'administration des environnements conteneurisés sur le système.

D-6.Architecture de Docker



1. Docker Client envoie des requête reçues pour construire une image (*docker build*), ou d'utilisation d'image existante est envoyée (*docker pull*, *docker run*).
2. Docker Daemon écoute les commandes envoyés par Docker Client et les exécute:
 - Dans le cas où l'image est **non-disponible localement**, alors le processus Docker va la **trouver en ligne** dans le registre configuré (par défaut docker.io).
 - Une fois l'image construite ou téléchargée, alors on continue vers l'étape 3.
 - Si l'image est **disponible localement**, alors on ignore le registre en ligne de docker et on va à l'étape 3.
 - Dans le cas de la construction, le contexte (l'arborescence de fichier au même niveau que le fichier «Dockerfile») est envoyé au processus.
3. L'image est **résolue** et **vérifiée** (comparaison de checksum), puis **stockée localement**, de façon accessible au processus docker.
4. Cette partie se produit **seulement** lors de la commande *docker run* : un **conteneur Docker** est **construit avec l'image de l'étape #2 et #3** puis est exécuté sur l'hôte, **indépendamment du système d'exploitation**.

D-7.Docker Compose

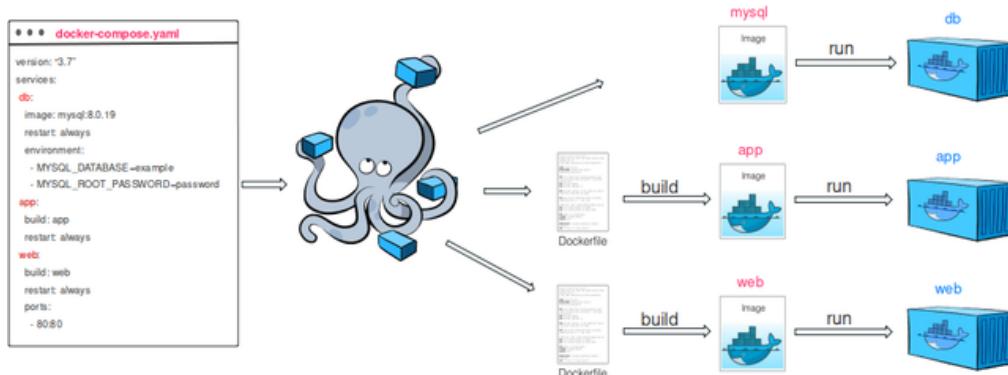
Docker Compose est un outil pratique pour **gérer plusieurs conteneurs Docker en tant que service unique**. Il simplifie le processus en permettant à ces conteneurs de fonctionner ensemble de manière cohérente. Voici comment nous pouvons l'utiliser en trois étapes principales :

1.Créer un Dockerfile : Commençons par créer un Dockerfile pour chaque composant de votre application. Ce fichier définit l'environnement nécessaire pour que l'application puisse être exécutée de manière reproductible.

2.Configuration des services dans *docker-compose.yml* : Ensuite, définissons chaque service nécessaire à notre application dans un fichier `docker-compose.yml`. Ce fichier spécifie **comment chaque conteneur doit être configuré** et comment ils **interagiront entre eux** dans un environnement isolé.

3.Démarrage avec Docker Compose : Utilisons la commande `docker-compose up` pour démarrer et exécuter tous les services définis dans notre fichier `docker-compose.yml`. Cela lance les conteneurs de manière coordonnée, en s'assurant qu'ils peuvent interagir les uns avec les autres selon les besoins.

Cet outil est particulièrement utile pour configurer et démarrer des services dépendants, comme des bases de données, des caches ou des interfaces API, de manière simple et fiable.



D-8.DockerFile

Un Dockerfile est un **fichier texte** qui contient des **instructions pour créer des images Docker**. Lorsque vous utilisez le fichier, Docker construit automatiquement une nouvelle image basée sur ces instructions. Voici comment vous pouvez procéder pour construire une image :

1. Commande de base : Commençons par entrer la commande `**docker build**`. Cela permet d'envoyer tous les fichiers du contexte, c'est-à-dire l'ensemble de fichiers dans un chemin ou URL spécifié, au démon Docker.

2. Spécification du Dockerfile : Si nous devons spécifier un Dockerfile qui n'est pas dans le répertoire actuel, utilisez la commande suivante :

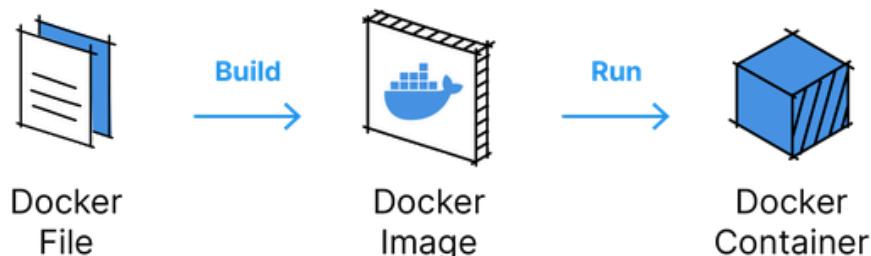
docker build -f /chemin/vers/le/Dockerfile

3. Taguer l'image : Une fois la construction réussie, vous pouvez taguer votre image pour l'enregistrer sous un nom spécifique dans un dépôt. Utilisez cette commande pour le faire :

docker build -t nomutilisateur/nomapp

4. Validation et erreurs : Pendant la construction, le démon Docker valide le Dockerfile. Si une erreur de syntaxe est détectée, une erreur sera renvoyée.

Ces étapes permettent de construire efficacement une image Docker en utilisant un Dockerfile.



E- Commandes de base

docker version

Pour vérifier la version de Docker installée

```
root@k-master:/home/osboxes# docker version
Client:
  Version:          19.03.6
  API version:      1.40
  Go version:       go1.12.17
  Git commit:       369ce74a3c
  Built:            Fri Feb 28 23:45:43 2020
  OS/Arch:          linux/amd64
  Experimental:    false

Server:
  Engine:
    Version:          19.03.6
    API version:      1.40 (minimum version 1.12)
    Go version:       go1.12.17
    Git commit:       369ce74a3c
    Built:            Wed Feb 19 01:06:16 2020
    OS/Arch:          linux/amd64
    Experimental:    false
  containerd:
    Version:          1.3.3-0ubuntu1~18.04.2
    GitCommit:
  runc:
    Version:          spec: 1.0.1-dev
    GitCommit:
  docker-init:
    Version:          0.18.0
    GitCommit:
```

docker info

Pour obtenir des informations détaillées sur votre configuration Docker

```
root@k-master:/home/osboxes# docker info
Client:
  Debug Mode: false

Server:
  Containers: 17
  Running: 16
  Paused: 0
  Stopped: 1
  Images: 11
  Server Version: 19.03.6
  Storage Driver: overlay2
    Backing Filesystem: extfs
    Supports d_type: true
    Native Overlay Diff: true
  Logging Driver: json-file
  Cgroup Driver: cgroupfs
  Plugins:
    Volume: local
    Network: bridge host ipvlan macvlan null overlay
      Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
  Swarm: inactive
  Runtimes: runc
  Default Runtime: runc
  Init Binary: docker-init
  containerd version:
  runc version:
  init version:
  Security Options:
    apparmor
    seccomp
      Profile: default
  Kernel Version: 5.0.0-23-generic
  Operating System: Ubuntu 18.04.3 LTS
github.com/kur
```

Format:

docker <command> <sub-command>

docker help

```
root@k-node02:/home/osboxes# docker help
Usage: docker [OPTIONS] COMMAND
A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default "/root/.docker")
  -c, --context string Name of the context to use to connect to the daemon (overrides DOCKER_HOST env var
                        context use)
  -D, --debug          Enable debug mode
  -H, --host list       Daemon socket(s) to connect to
  -l, --log-level string Set the logging level ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls                Use TLS; implied by --tlsv1.2
  --tlscacert string   Trust certs signed only by this CA (default "/root/.docker/ca.pem")
  --tlscert string     Path to TLS certificate file (default "/root/.docker/cert.pem")
  --tlskey string       Path to TLS key file (default "/root/.docker/key.pem")
  --tlsv1.2              Use TLS and verify the remote
  -v, --version         Print version information and quit

Management Commands:
  builder    Manage builds
  config     Manage Docker configs
  container  Manage containers
  context    Manage contexts
  engine     Manage the docker engine
  image      Manage images
  network   Manage networks
  node       Manage Swarm nodes
  plugin    Manage plugins
  secret    Manage Docker secrets
  service   Manage services
  stack     Manage Docker stacks
```

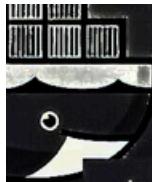
docker <command> <sub-command>

docker image --help

```
root@k-node02:/home/osboxes# docker image --help
Usage: docker image COMMAND

Manage images

Commands:
  build      Build an image from a Dockerfile
  history    Show the history of an image
  import     Import the contents from a tarball to create a filesystem image
  inspect    Display detailed information on one or more images
  load       Load an image from a tar archive or STDIN
  ls        List images
  prune     Remove unused images
  pull       Pull an image or a repository from a registry
  push       Push an image or a repository to a registry
  rm        Remove one or more images
  save      Save one or more images to a tar archive (streamed to STDOUT by default)
  tag       Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
```



ESSENTIAL DOCKER COMMANDS

<code>docker run</code>	Create a container and start it immediately
<code>docker stop</code>	Stop containers that are already running
<code>docker start</code>	Start containers that have been stopped
<code>docker exec</code>	Execution of new commands within a container that's already running
<code>docker rename</code>	Rename a container
<code>docker pause/unpause</code>	Suspend all processes in a specified container
<code>docker kill</code>	Send the default KILL signal to a container
<code>docker build</code>	Build modified (custom) images
<code>docker cp</code>	Copy files and directories between the container and host.
<code>docker rm</code>	Remove both images and containers.
<code>docker ps/container ls</code>	List all the containers that are currently running on a system
<code>docker images/image ls</code>	Check the list of all Docker images existing within a system
<code>docker logs</code>	Fetch the logs of any container
<code>docker version</code>	Obtain the detailed information on the Docker version installed
<code>docker info</code>	Show system wide information on the host system where Docker is installed
<code>docker inspect</code>	Reveal low-level information about both containers and images
<code>docker history</code>	History of any image existing on your Docker system
<code>docker port</code>	Reveals host to container port mappings
<code>docker diff</code>	Inspect changes to files or directories on a container's file system
<code>docker top</code>	List the processes running within a container
<code>docker stats</code>	Shows container details along-with system resource usage

LINUXHUBBOOK

F- Mise en Pratique : Installation et Configuration de Docker

F-1.Installer Docker

1-Mise à jour du système :

```
> sudo apt update  
> sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

2-Installation des prérequis : Installons les paquets nécessaires pour permettre à apt de gérer les paquets via HTTPS.

Ajout de la clé GPG du dépôt Docker : Téléchargeons et ajoutons la clé GPG de Docker à votre système.

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

3-Ajout du dépôt Docker : Configurons les sources APT pour inclure le dépôt Docker.

```
> 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
```

4-Installation de Docker : mettons à jour des packets de nouveau, installons à partir du Repo Docker

```
> 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
```

5-Test de l'installation : Vérifions le bon fonctionnement

```
> docker run hello-world
```

Astuce: Si une erreur de permissions survient

```
yaya@yayafirdaous:~$ docker run hello-world  
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock  
See 'docker run --help'.
```

Cette erreur indique que vous n'avez pas les autorisations nécessaires pour vous connecter au socket du démon Docker

Pour la résoudre il suffit de suivre ces étapes pour ajouter notre utilisateur au groupe Docker et réessayons .

Pour la résoudre il suffit de suivre ces étapes pour ajouter notre utilisateur au groupe Docker et réessayons .

```
> sudo groupadd docker  
> sudo usermod -aG docker $USER  
> newgrp docker
```

```
aya@ayaafirdaous:~$ sudo groupadd docker  
aya@ayaafirdaous:~$ sudo usermod -aG docker $USER  
aya@ayaafirdaous:~$ newgrp docker
```

Réessayons :

```
aya@ayaafirdaous:~$ docker run hello-world  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
(amd64)  
3. The Docker daemon created a new container from that image which runs the  
executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

Pour vérifier que l'image est bien créé:

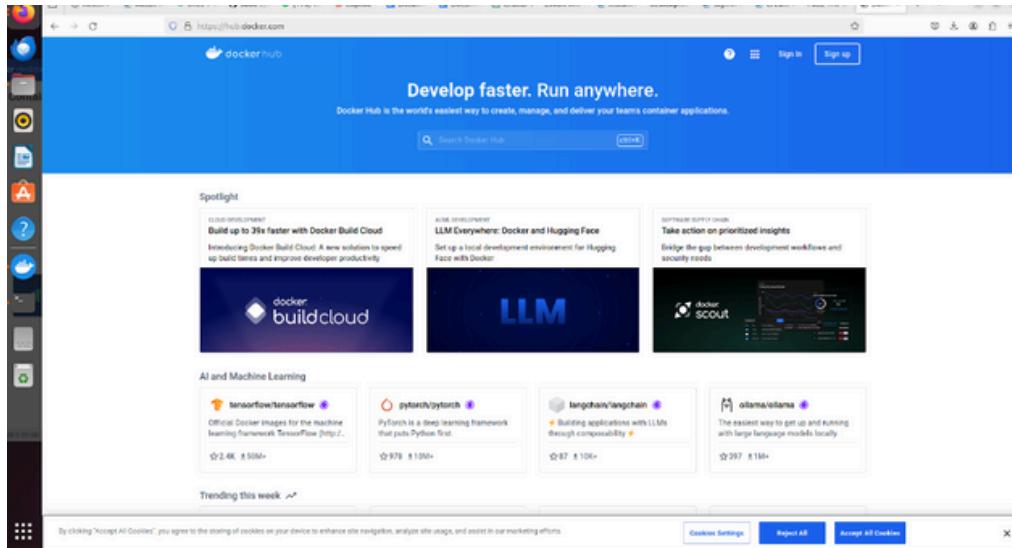
```
> docker ps -a
```

```
aya@ayaafirdaous:~$ docker ps -a  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS          NAMES  
07a8be7c7337        hello-world        "/hello"           2 hours ago       Exited (0) 2 hours ago   modest_ellis  
b995a53d5350        hello-world        "/hello"           2 hours ago       Exited (0) 2 hours ago   elegant_lewin
```

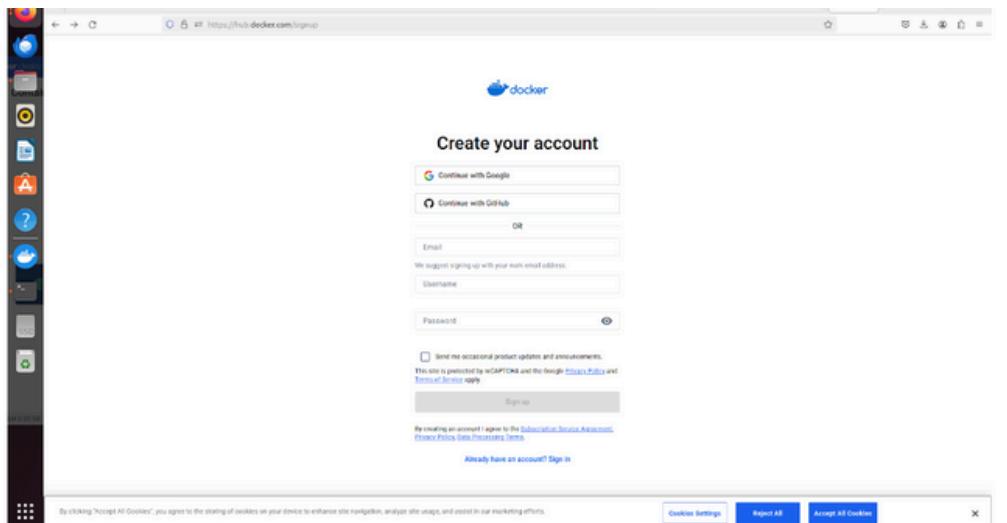
F-2.Créer un compte sur Docker Hub

Pour commencer à utiliser Docker Hub et gérer vos images Docker, suivez ces étapes simples :

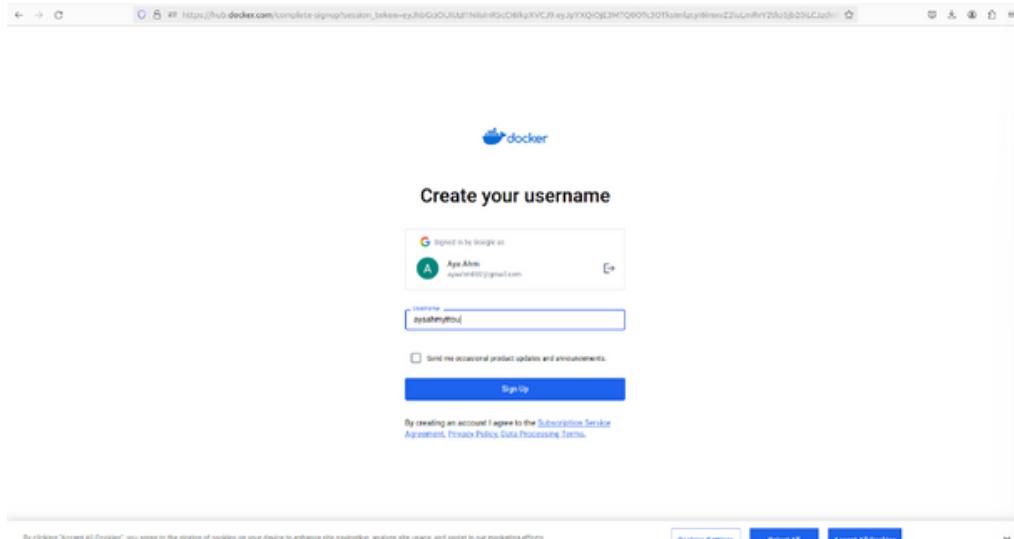
1-Rendons-nous sur le site officiel de <https://hub.docker.com/>



2-Inscription : Cliquer sur le bouton '**Sign Up**' pour démarrer le processus d'inscription. Remplir les champs requis avec les informations personnelles.



3-Choix d'un Identifiant : Sélectionnez un nom d'utilisateur unique qui sera votre identifiant sur Docker Hub.



4-Confirmation et Utilisation : Une fois l'inscription terminée, vous pouvez commencer à utiliser Docker Hub pour gérer vos images Docker.

F-3.Installer Docker Desktop

1-Téléchargement de Docker Desktop : Accédez à la page de téléchargement officielle de Docker Desktop et téléchargez le paquet approprié pour Linux à l'adresse suivante

:[https://desktop.docker.com/linux/main/amd64/145265/docker-desktop-4.29.0-amd64.deb?](https://desktop.docker.com/linux/main/amd64/145265/docker-desktop-4.29.0-amd64.deb?utm_source=docker&utm_medium=webreferral&utm_campaign=docs-driven-download-linux-amd64&_gl=1*1gr2sk9*_ga*MTU3MzM0MzM3LjE3MTQ0Nzc4Mjc.*_ga_XJWPQMJYHQ*MTcxNDQ5NDc5My4zLjEuMTcxNDQ5NDgzNy4xNi4wLjA)

2-Installation du logiciel :

- Naviguez vers le répertoire de téléchargement, généralement ~/Downloads, et vérifiez la présence du fichier téléchargé.

```
>cd Downloads/  
> ll
```

```
aya@ayafirdaous:~$ cd Downloads/  
aya@ayafirdaous:~/Downloads$ ll  
total 547796  
drwxr-xr-x 2 aya aya 4096 17:39 30 . /  
drwxr-x--- 19 aya aya 4096 12:40 30 .. /  
-rw-rw-r-- 1 aya aya 101661098 19:31 21 code_1.88.1-1712771838_amd64.deb  
-rw-rw-r-- 1 aya aya 343238956 14:09 30 docker-desktop-4.29.0-amd64.deb  
-rw-rw-r-- 1 aya aya 106978272 09:29 24 google-chrome-stable_current_amd64.deb  
-rw-rw-r-- 1 aya aya 6488814 15:04 30 'LINUX-RAPPORT TECHNIQUE.pdf'  
-rw-rw-r-- 1 aya aya 220171 14:05 24 'Série de TP N°1_2024.pdf'  
-rw-rw-r-- 1 aya aya 1133430 09:31 24 'Support cours Economie 1- 2024.pdf'  
-rw-rw-r-- 1 aya aya 163275 09:17 24 TD2_Cpp_Heritage.pdf
```

- Ouvrons un terminal et exécutons les commandes suivantes pour mettre à jour notre système et installer Docker Desktop :

```
>sudo apt-get update  
>sudo apt-get install ./docker-desktop-4.29.0-amd64.deb
```

```
aya@ayafirdaous:~/Downloads$ sudo apt-get install ./docker-desktop-4.29.0-amd64.deb  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Note, selecting 'docker-desktop' instead of './docker-desktop-4.29.0-amd64.deb'  
The following packages were automatically installed and are no longer required:  
  bridge-utils ubuntu-fan  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  cpu-checker ibverbs-providers ipxe-qemu ipxe-qemu-256k-compat-efi-roms libaio1 libcacard0 libdaxctl1  
  libdecor-0-0 libdecor-0-plugin-1-cairo libfdt1 libgfapi0 libgfrpc0 libgfxdr0 libglusterfs0  
  libibverbs1 liblscsi1 libndctl6 libpmem1 libpmemobj1 libqrencode4 librados2 librbd1 librdmacm1  
  libSDL2-2.0-0 libspice-server1 liburing2 libusbredirparser1 libvirglrenderer1 msr-tools ovmf pass  
  
Setting up qemu-system-gui (1:6.2+dfsg-2ubuntu6.19) ...  
Setting up libgfapi0:amd64 (10.1-1ubuntu0.2) ...  
Setting up qemu-block-extra (1:6.2+dfsg-2ubuntu6.19) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/run-qemu.mount → /lib/systemd/system/run-qemu.  
.mount.  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...  
Processing triggers for libc-bin (2.35-0ubuntu3.7) ...  
Processing triggers for man-db (2.10.2-1) ...  
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...  
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...  
N: Download is performed unsandboxed as root as file '/home/aya/Downloads/docker-desktop-4.29.0-amd64.deb'  
' couldn't be accessed by user 'apt'. - pkgAcquire::Run (13: Permission denied)
```

- Si un message d'erreur de permission apparaît, il peut être ignoré pour poursuivre.

3-Démarrage de Docker Desktop :

- Lancer Docker Desktop:

```
>systemctl --user start docker-desktop
```

```

aya@ayafiredaous:~$ sudo apt update
aya@ayafiredaous:~$ sudo apt upgrade
aya@ayafiredaous:~$ 
```

Docker Subscription Service Agreement

By selecting accept, you agree to the [Subscription Service Agreement](#), the [Docker Data Processing Agreement](#), and the [Data Privacy Policy](#).

Note: Docker Desktop is free for small businesses (fewer than 250 employees AND less than \$10 million in annual revenue), personal use, education, and non-commercial open source projects. Otherwise, it requires a paid subscription for professional use. Paid subscriptions are also required for government entities.

Read the [FAQ](#) to learn more.

[View Full Terms](#) [Accept](#) [Close](#)

- Suivre les étapes
- Si une erreur survient lors de la connexion, générer une nouvelle clé GPG et initialiser avec pass pour résoudre le problème.

>gpg --generate-key

```

aya@ayafiredaous:~$ gpg --generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg --full-generate-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: ayaahmyttou
Email address: ayaahm002@gmail.com
You selected this USER-ID:
  "ayaahmyttou <ayaahm002@gmail.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key 310EF5C67DCA938B marked as ultimately trusted
gpg: revocation certificate stored as '/home/aya/.gnupg/openpgp-revocs.d/3ADB90600D0256AEA52D04E7310EF5C67DCA938B.rev'
public and secret key created and signed.

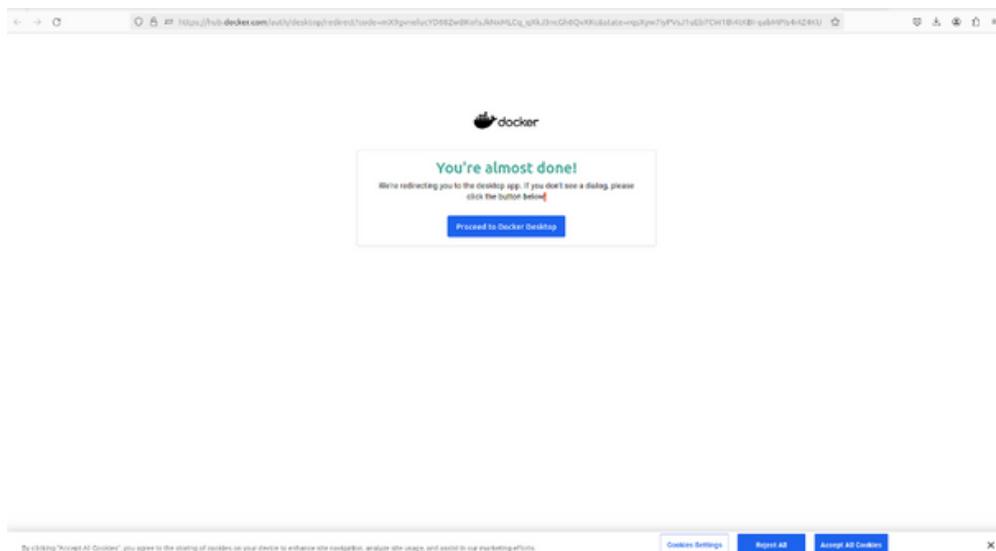
pub    rsa3072 2024-04-30 [SC] [expires: 2026-04-30]
      3ADB90600D0256AEA52D04E7310EF5C67DCA938B
uid    ayaahmyttou <ayaahm002@gmail.com>
sub    rsa3072 2024-04-30 [E] [expires: 2026-04-30]
aya@ayafiredaous:~$ 
```

```
>pass init 3ADB90600D0256AEA52D04E7310EF5C67DCA938B
```

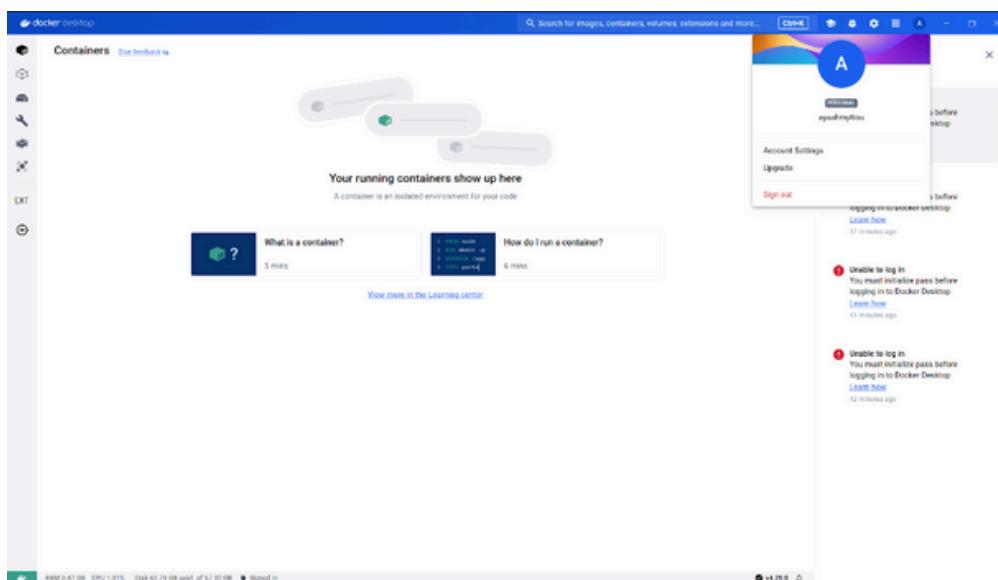
```
ous:~$ pass init 3ADB90600D0256AEA52D04E7310EF5C67DCA938B
ed directory '/home/aya/.password-store/'
re initialized for 3ADB90600D0256AEA52D04E7310EF5C67DCA938B
```

4- Connexion et utilisation :

- Une fois Docker Desktop lancé, suivre les instructions à l'écran pour se connecter à notre compte Docker ou en créer un nouveau si nécessaire.



- Après la connexion, nous serons redirigé vers l'interface principale de Docker Desktop où nous pouvons gérer nos conteneurs et images Docker.



F-4.Créer la structure du projet:

1-Préparation de l'Espace de Travail :

- Création du Dossier Racine : Commencer par créer un dossier principal pour le projet, qui hébergera tous les composants nécessaires.
- Sous-dossiers pour le Backend et le Frontend : Dans ce dossier racine, créez deux sous-dossiers distincts pour le backend et le frontend. Cela aide à organiser clairement les différentes parties de l'application.
-

2-Configuration des dossiers :

- Backend : Le dossier backend contiendra tous les fichiers spécifiques au backend de l'application, utilisant Laravel.
- Frontend : Le dossier frontend inclura tous les fichiers liés à l'interface utilisateur, développés avec Vue.js.

3-Fichiers de Configuration Docker :

- Dockerfile pour le Backend et le Frontend : Chaque sous-dossier doit contenir un Dockerfile qui définit les instructions pour construire les images Docker pour le backend et le frontend respectivement.
- Fichier docker-compose.yml : À la racine du projet, ajoutez un fichier docker-compose.yml. Ce fichier coordonne tous les services Docker, facilitant leur gestion comme une seule application.

4-Configuration de NGINX :

- Fichier nginx.conf : Préparez un fichier de configuration NGINX (nginx.conf) à la racine du projet. Ce fichier détermine comment les requêtes sont routées vers le backend ou le frontend selon l'URL.

Cette structure organisée garantit que tous les éléments du projet sont bien rangés et facilement accessibles, optimisant le développement et les déploiements ultérieurs.

F-5.Mise en Place de l'Environnement de Développement avec Docker pour Laravel

1-Préparation de l'Espace de Travail :

- Création du Dossier Principal : Ouvrez un terminal et créez un dossier principal pour votre projet, par exemple ~/inno_kids :

```
>mkdir ~/inno_kids  
>cd ~/inno_kids
```

- Configuration du Backend : À l'intérieur de ce dossier principal, créez un sous-dossier backend et naviguez dedans :

```
>mkdir backend  
>cd backend
```

2-Installation de Laravel via Composer :

```
>sudo apt install php8.1-curl //Installation de l'extension CURL pour PHP  
>sudo service nginx restart //Redémarrage du service Nginx
```

```
aya@ayafirdaous:~/inno_kids$ sudo apt install php8.1-curl  
[sudo] password for aya:  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
php8.1-curl is already the newest version (8.1.2-1ubuntu2.17).  
The following packages were automatically installed and are no longer required:  
  bridge-utils ubuntu-fan  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 35 not upgraded.  
aya@ayafirdaous:~/inno_kids$ sudo service nginx restart
```

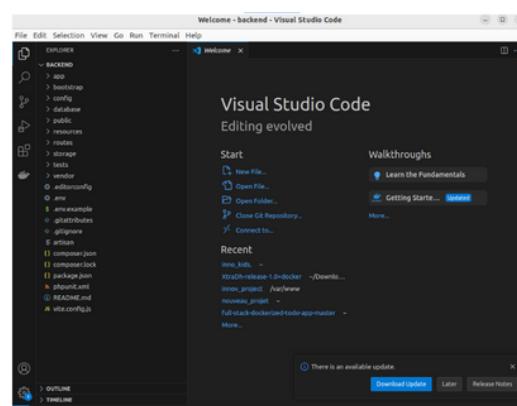
- Préparation de Laravel : Utilisez Composer pour installer Laravel directement dans le sous-dossier backend avec la commande :

```
>composer create-project --prefer-dist laravel/laravel .
```

```
aya@ayafirdaous:~/inno_kids/backend$ composer create-project --prefer-dist laravel/laravel .
```

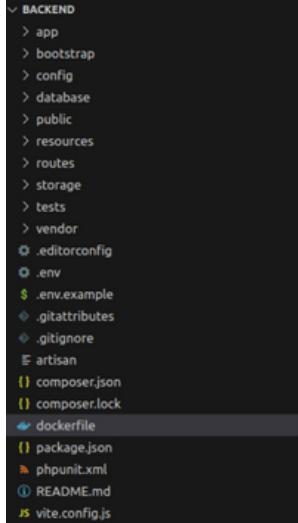
- Ouvrir le projet dans Visual Studio Code :

```
>code .
```



3-Configuration de Docker :

- Création d'un Dockerfile : Dans le répertoire du backend, créer un Dockerfile qui contiendra toutes les directives nécessaires à la construction de l'image Docker pour Laravel.



```

# Welcome
# dockerfile x
# dockerfile > ...
1 # 2 - The Laravel API :
2 FROM php:8.1-fpm
3
4 # Mise à jour des paquets et installation des dépendances
5 RUN apt-get update && apt-get install -y \
6     libpq-dev \
7     git \
8     unzip \
9     libonig-dev \
10    libxml2-dev \
11    zip \
12    && docker-php-ext-install pdo pdo_pgsql
13
14 # Installer Composer dans une étape séparée
15 COPY --from=composer:latest /usr/bin/composer /usr/local/bin/composer
16
17
18 #RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer
19
20 # Définir le répertoire de travail
21 WORKDIR /var/www/api
22
23 # Copier l'application Laravel dans l'image
24 COPY . /var/www/api
25
26 # Permettre à Composer de s'exécuter en tant que super utilisateur
27 ENV COMPOSER_ALLOW_SUPERUSER=1
28
29 # Installer les dépendances PHP avec Composer
30 RUN composer install --optimize-autoloader --no-dev
31
32 # Générer la clé d'application Laravel (à faire hors de l'image pour la sécurité)
33 RUN php artisan key:generate
34
35 #Modifier les permissions pour les répertoires de cache et de stockage
36 RUN chown -R www-data:www-data /var/www/api/storage /var/www/api/bootstrap/cache
37
38 #CMD php artisan serve --host='0.0.0.0' --port=8000
39 CMD ["php-fpm"]

```

- Construction de l'Image Docker : Construisez l'image en exécutant docker build -t backend . dans le terminal.

```
>docker build -t backend .
```

- Vérifier si l'image Docker a été correctement créée

```
>docker images
```

```

REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
backend            latest   ca1b03c19db2  13 minutes ago  680MB
laravel_1          latest   7cf236f13438  7 days ago    680MB
nom_de_votre_image latest   d0894abfbe8b  9 days ago    671MB
larael_test_1      latest   b4e00993c0bf  10 days ago   671MB
mysql              5.7     5107333e08a8  5 months ago   501MB

```

4-Lancement et Validation :

- Démarrage du Conteneur Docker : Lancer le conteneur basé sur l'image créée

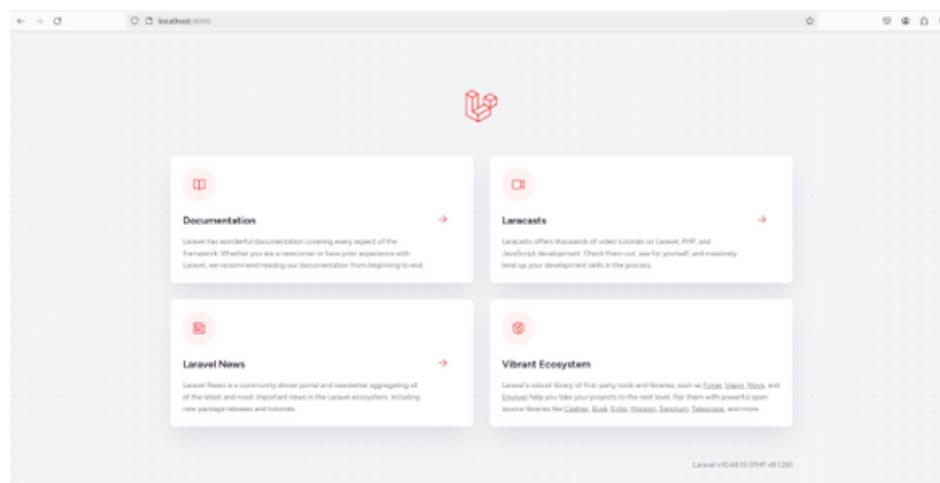
```
>docker ps
```

- Vérification de Fonctionnement : S'assurer que le conteneur fonctionne correctement en vérifiant son statut avec docker ps

```
>docker run -d -p 8000:8000 backend
```

```
● aya@ayafirdaous:~/inno_kids/backend$ docker run -d -p 8000:8000 backend
9c71e7f5aaca38cd1ab59c97754fa8dbf18af8af329c1cbc77a4ce89b88
● aya@ayafirdaous:~/inno_kids/backend$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
9c71e7f5aaca        backend             "docker-php-entrypoi..."   15 seconds ago    Up 15 seconds          0.0.0.0:8000->8000/tcp, 9000/tcp   jovial_napier
```

- et en accédant à Laravel via: <http://localhost:8000/>.



5-Publication sur Docker Hub :

- Connexion à Docker Hub

```
>docker login
```

- Taguer l'image Docker avec
- > docker tag backend ayaahmyttou/backend

- publier-la sur Docker Hub avec

```
• > docker push ayaahmyttou/backend
```

```

● aya@ayafirdaous:~/inno_kids/backend$ docker login
Authenticating with existing credentials...
Login Succeeded
● aya@ayafirdaous:~/inno_kids/backend$ docker tag backend ayaahmyttou/backend
● aya@ayafirdaous:~/inno_kids/backend$ dockerpush ayaahmyttou/backend
dockerpush: command not found
● aya@ayafirdaous:~/inno_kids/backend$ docker push ayaahmyttou/backend
Using default tag: latest
The push refers to repository [docker.io/ayaahmyttou/backend]
dbdd4783ee0c: Pushed
545e59b7b3c6: Pushed
042b208af27: Pushed

```

Local Hub

ayaahmyttou
 Search
[View Scout dashboard](#)

	Tags	OS	Vulnerabilities	Last pushed	Size
 ayaahmyttou/backend	latest		Inactive	54 seconds ago	253.19 M

F-5.Mise en Place de l'Environnement de Développement avec Docker pour VueJs et Nginx

1-Installation des Prérequis :

- Node.js et npm : Installez Node.js et npm, qui sont essentiels pour le développement avec Vue.js.

```
> curl -fsSL https://deb.nodesource.com/setup_current.x | sudo -E bash - //installer
npm
```

```

curl: (28) Failed to open file or directory
aya@ayafirdaous:~/inno_kids/frontend$ curl -fsSL https://deb.nodesource.com/setup_current.x | sudo -E bash -
2024-05-12 22:24:44 - Installing pre-requisites
Hit:1 http://ma.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 https://download.docker.com/linux/ubuntu jammy InRelease
Get:3 https://packages.microsoft.com/repos/stable InRelease [3,590 B]
Get:4 http://ma.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:6 https://ppa.launchpadcontent.net/git-core/ppa/ubuntu jammy InRelease
Hit:7 http://ma.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 233 kB in 1s (219 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.16).
unzip is already the newest version (2.2.27-3ubuntu2.1).

```

- pour s'assurer que c'est bien installé

```
> npm -v
>node -v
```

```

aya@ayafirdaous:~/inno_kids/frontend$ node -v
v22.1.0
aya@ayafirdaous:~/inno_kids/frontend$ npm -v
10.7.0

```

2-Installation de Vue CLI :

- Vue CLI : Installez Vue CLI globalement en utilisant npm. Cette interface de ligne de commande est nécessaire pour créer et gérer les projets Vue.js.

```
> sudo npm install -g @vue/cli
```

```
aya@ayafirdaous:~/inno_kids/frontend$ sudo npm install -g @vue/cli
npm warn EBAENGINE Unsupported engine {
  npm warn   EBAENGINE   package: '@achirmaza/node-ipc@0.2.8',
  npm warn   EBAENGINE   required: {
  npm warn     EBAENGINE     node: '8 || 9 || 10 || 11 || 12 || 13 || 14 || 15 || 16 || 17 || 18 || 19 || 20 || 21'
  npm warn   EBAENGINE   },
  npm warn   EBAENGINE   current: { node: 'v22.1.0', npm: '10.7.0' }
  npm warn   EBAENGINE }

npm warn deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm warn deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm warn deprecated urix@0.1.0: See https://github.com/lydell/urix#deprecated
npm warn deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm warn deprecated babel-plugin-proposal-nullish-coalescing-operator@7.18.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.
npm warn deprecated babel-plugin-proposal-optional-chaining@7.11.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm warn deprecated babel-plugin-proposal-class-properties@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.
npm warn deprecated apollo-server-plugin-base@3.7.7: The 'apollo-server-plugin-base' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated apollo-server-errors@3.3.1: The 'apollo-server-errors' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated apollo-data-source@3.3.2: The 'apollo-data-source' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023 and October 22nd 2024, respectively). See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated apollo-reporting-protobuf@3.4.0: The 'apollo-reporting-protobuf' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/usage-reporting-protobuf' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated apollo-server-env@4.2.1: The 'apollo-server-env' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/utils-fetcher' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated apollo-server-types@3.8.0: The 'apollo-server-types' package is part of Apollo Server v2 and v3, which are now deprecated (end-of-life October 22nd 2023 and October 22nd 2024, respectively). This package's functionality is now found in the '@apollo/server' package. See https://www.apollographql.com/docs/apollo-server/previous-versions/ for more details.
npm warn deprecated subscriptions-transport-ws@0.11.0: The 'subscriptions-transport-ws' package is no longer maintained. We recommend you use 'graphql-ws' instead. For help migrating Apollo software to 'graphql-ws', see https://www.apollographql.com/docs/apollo-server/data/subscriptions/#switching-from-subscriptions-transport-ws. For general help using graphql-ws, see https://github.com/enisimov/graphql-ws/blob/master/README.md
npm warn deprecated shortid@0.2.16: Package no longer supported. Contact Support at https://www.npmjs.com/support for more info.
npm warn deprecated vue@2.7.16: Vue 2 has reached EOL and is no longer actively maintained. See https://v2.vuejs.org/es/ for more details.

added 853 packages in 2s

74 packages are looking for funding
```

- pour s'assurer que c'est bien installé

```
> vue --version
```

```
aya@ayafirdaous:~/inno_kids/frontend$ vue --version
@vue/cli 5.0.8
```

3-Création du Projet Vue.js :

- Initialisation du Projet : Utiliser Vue CLI pour créer un nouveau projet Vue.js. Se positionner dans le dossier où vous souhaitez créer le projet et exécuter :

```
> vue create inno_kids
```

```
aya@ayafirdaous:~/inno_kids/frontend$ vue create .
(node:43456) [DEP0040] DeprecationWarning: The 'punycode' module is deprecated. Please use a userland alternative instead.
(Use 'node --trace-deprecation ...' to show where the warning was created)

Vue CLI v5.0.8
? Generate project in current directory? Yes

Vue CLI v5.0.8
? Please pick a preset: Default ([Vue 3] babel, eslint)

Vue CLI v5.0.8
  🌟 Creating project in /home/aya/inno_kids/frontend.
  🌟 Initializing git repository...
  🌟 Installing CLI plugins. This might take a while...

added 872 packages, and audited 873 packages in 1m
```

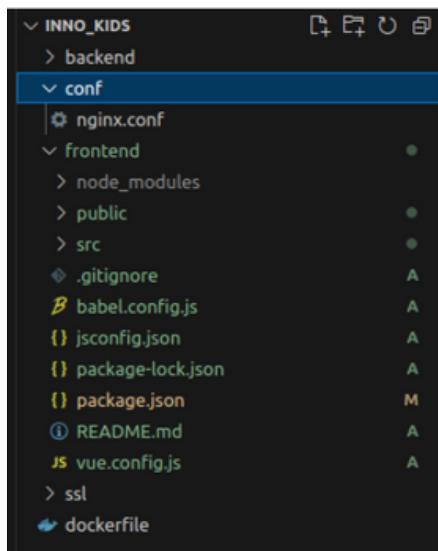
- Démarrage du Serveur de Développement : Une fois le projet créé, lancez le serveur de développement pour accéder à l'application Vue.js localement.

```
>cd inno_kids
>npm run serve
```

```
App running at:
- Local: http://localhost:8080/
- Network: http://192.168.1.119:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

4-Compléter la structure du projet InnoKids:



- NB: Ce dockerfile qui est dans le dossier du projet qui définit comment construire l'image Docker pour Nginx+VueJs (la justification sera dans le paragraphe dessus)

5-Configuration de Docker pour Vue.js :

- Dockerfile pour Vue.js : Créez un Dockerfile dans le dossier du projet qui définit comment construire l'image Docker pour Vue.js. et Nginx:

```

# - 1 - The VueJs Application :
FROM node:19 as builder

WORKDIR /vue-ui

# Corrigez le chemin ici pour pointer vers le bon répertoire
COPY frontend/package*.json .

RUN npm install

# Assurez-vous que le chemin ici est également correct
COPY frontend/.. .

RUN npm run build

# - 3 - The NGINX Server :
FROM nginx:latest

COPY conf/nginx.conf /etc/nginx/nginx.conf
COPY ssl /usr/share/nginx/ssl

RUN rm -rf /usr/share/nginx/html/*
COPY --from=builder /vue-ui/dist /usr/share/nginx/html

EXPOSE 80
ENTRYPOINT ["nginx", "-g", "daemon off;"]

```

- Construction et Lancement de l'Image Docker : Construire l'image Docker et lancer le conteneur.

```
> docker build -t frontinx .
> docker run -d -p 8080:8080 vuejs_app
```

```
aya@ayafirdaous:~/inno_kids$ docker build -t frontinx .
[+] Building 83.0s (19/19) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 681B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load metadata for docker.io/library/node:19
=> [auth] library/nginx:pull token for registry-1.docker.io
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [builder 1/6] FROM docker.io/library/node:19@sha256:92f06fc13bcc09f1ddc51f6e
=> [internal] load build context
=> => transferring context: 2.14MB
=> [stage-1 1/5] FROM docker.io/library/nginx:latest@sha256:32e76d4f34f80e47996
=> CACHED [builder 2/6] WORKDIR /vue-ui
=> CACHED [stage-1 2/5] COPY conf/nginx.conf /etc/nginx/nginx.conf
=> CACHED [stage-1 3/5] COPY ssl /usr/share/nginx/ssl
=> CACHED [stage-1 4/5] RUN rm -rf /usr/share/nginx/html/*
=> [builder 3/6] COPY frontend/package*.json .
=> [builder 4/6] RUN npm install
=> [builder 5/6] COPY frontend/..
=> [builder 6/6] RUN npm run build
=> [stage-1 5/5] COPY --from=builder /vue-ui/dist /usr/share/nginx/html
=> exporting to image
=> => exporting layers
```

```
> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
frontinx	latest	51d8fd75421c	25 seconds ago	189MB
ayaahmyttou/backend	latest	9ec63348721c	20 hours ago	680MB
backend	latest	9ec63348721c	20 hours ago	680MB

6-Déploiement sur Docker Hub :

- Taggage et Publication : Tagger l'image Docker créée et publier-la sur Docker Hub pour permettre son déploiement et son partage.

```
> docker tag vuejs_app username/vuejs_app
> docker push username/vuejs_ap
```

```
aya@ayafirdaous:~/inno_kids$ docker tag frontinx:latest ayaahmyttou/inno_kids:frontinx
aya@ayafirdaous:~/inno_kids$ docker push ayaahmyttou/inno_kids:frontinx
The push refers to repository [docker.io/ayaahmyttou/inno_kids]
6f794a453fe3: Pushed
28325c92934f: Pushed
ef85a6d5147c: Pushed
e7ea734c38c5: Pushed
b089746990db: Mounted from library/nginx
3f69a32a2b31: Mounted from library/nginx
d75fd41110a4: Mounted from library/nginx
2ebd92876e9f: Mounted from library/nginx
a8a738834a4d: Mounted from library/nginx
bf7cd01dee5e: Mounted from library/nginx
52e5a4310fa: Mounted from ayaahmyttou/backend
frontinx: digest: sha256:9015b4e0886401838b2887b9decc78e196b6f089d9f3b4d4428dec84ece4cdd0 size: 2609
aya@ayafirdaous:~/inno_kids$ docker tag backend:latest ayaahmyttou/inno_kids:backend
aya@ayafirdaous:~/inno_kids$ docker push ayaahmyttou/inno_kids:backend
The push refers to repository [docker.io/ayaahmyttou/inno_kids]
99ac7792aa68: Mounted from ayaahmyttou/backend
7b2ccc9322d6: Mounted from ayaahmyttou/backend
7eaefaf712a1: Mounted from ayaahmyttou/backend
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
cd3fb10edbb	backend	"docker-php-entrypoint"	9 minutes ago	Up 9 minutes	0.0.0.0:8000->8000/tcp, 9000/tcp	eplc_khorana
a2a702d5003b	frontinx	"nginx -g 'daemon off;'	21 minutes ago	Up 9 minutes	0.0.0.0:8080->80/tcp	beautiful_cham
42b26757c395	postgres:15	"docker-entrypoint.s..."	36 minutes ago	Up 10 minutes	0.0.0.0:5432->5432/tcp	db

Comparaison entre l'hébergement de Vue.js et Nginx dans un conteneur unique et dans des conteneurs séparés



Lors de l'architecture des applications web modernes utilisant Vue.js et Nginx, une question courante est de savoir si placer ces deux technologies dans un conteneur unique ou dans des conteneurs séparés est la meilleure pratique. Pour Vue.js, en particulier avec des serveurs de développement comme Vite ou Nuxt, regrouper Nginx et Vue.js **dans un seul conteneur peut s'avérer avantageux**.

D'un point de vue pratique, maintenir Vue.js et Nginx dans le même conteneur **réduit la latence et simplifie la communication interne**, puisque **les requêtes entre Vue.js et Nginx n'ont pas à traverser les réseaux de Docker**. Ceci est crucial pour **les applications nécessitant une réponse rapide et un minimum de délai**, comme celles **utilisant des interfaces utilisateur interactives et dynamiques**.

En revanche, pour des frameworks comme Laravel, qui sont souvent utilisés pour des applications nécessitant une architecture robuste avec des services backend complexes, il peut être préférable de séparer les conteneurs. Cette séparation facilite la scalabilité, la maintenance, et renforce la sécurité, chaque service s'exécutant de manière isolée.

Ainsi, pour Vue.js, l'intégration de Nginx dans le même conteneur offre des avantages en termes de performance et de simplicité de déploiement, tandis que pour des applications Laravel, la séparation entre Nginx et l'application elle-même est souvent recommandée pour optimiser la gestion des environnements de production.

F-7.Configuration de Docker Compose

1-Configuration de Docker Compose :

- Fichier docker-compose.yml : Ce fichier est crucial pour définir et configurer les services nécessaires à votre application. Il inclut des configurations pour les services backend, frontend et base de données.
 - Backend : Configurez le service backend avec un Dockerfile personnalisé et spécifiez les dépendances nécessaires.
 - Frontend : Définissez le service frontend qui dépend du backend pour fonctionner correctement.
 - Base de données : Utilisez une image PostgreSQL dont l'image est extraite directement de Docker Hub, configurez les ports et définissez les volumes pour la persistance des données.

```
version: "3"

services:
  backend:
    container_name: epic_khorana
    image: backend
    build:
      context: backend
      dockerfile: ./dockerfile
    env_file:
      - backend/.env
    depends_on:
      - db
    networks:
      - app-network
    ports:
      - "8000:8000"

  frontend:
    container_name: beautiful_chaur
    image: frontinx
    build:
      context: .
    ports:
      - "8080:80"
    depends_on:
      - db
    networks:
      - app_network

  db:
    container_name: db
    image: postgres:15
    restart: always
    ports:
      - 5432:5432
    volumes:
      - ./db/init.sql:/docker-entrypoint-initdb.d/0001-init.sql
      - ./db/triggers.sql:/docker-entrypoint-initdb.d/0002-init.sql
      - ./db/insertion.sql:/docker-entrypoint-initdb.d/0003-init.sql
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
      - POSTGRES_DB=postgres
      - POSTGRES_HOST=db
    networks:
      - app-network
```

2-Lancement des Services :

- Démarrer tous les services définis dans le fichier docker-compose.yml en mode détaché, permettant à notre application de s'exécuter en arrière-plan.

```
> docker compose up
```

Vérification et Gestion des Services :

- État des Conteneurs : Après le lancement des services, vérifiez l'état des conteneurs :

```
> docker -compose ps
```

Cette commande vous montre l'état de chaque conteneur, y compris les ports utilisés et le statut de fonctionnement.

- Logs des Conteneurs : Pour un diagnostic ou une vérification supplémentaire, consultez les logs des conteneurs :

```
> docker-compose logs
```

```
aya@ayaflrdaous:~/inno_kids$ docker-compose up -d
db is up-to-date
Creating beautiful_chaum ... done
Creating epic_khorana ... done
aya@ayaflrdaous:~/inno_kids$ docker-compose ps
          Name        Command     State            Ports
----- 
beautiful_chaum    nginx -g daemon off;
db                  docker-entrypoint.sh postgres Up      0.0.0.0:8080->80/tcp
epic_khorana       docker-php-entrypoint /bin ... Up      0.0.0.0:5432->5432/tcp
                                         9000/tcp
```

F-8.Gestion des Images Docker :

- Connexion à Docker Hub : Connectez-vous à votre compte Docker Hub avec la commande docker login.

```
> docker login
```

- Taggage de l'Image : Taggez votre image Docker pour la préparer à la publication avec docker tag.

```
>docker tag postgres:15 ayaahmyttou/inno_kids:postgres
```

- Publication de l'Image : Publiez l'image sur Docker Hub en utilisant docker push. Ceci permet de partager votre image avec d'autres ou de la déployer sur d'autres environnements ou services.

```
>docker push ayaahmyttou/inno_kids:postgres
```

```

username: ayaahmyttou
password: 
Login Succeeded
ayaahmyttou@Inno_Kids:~$ docker tag postgres:15 ayaahmyttou/inno_kids:postgres
ayaahmyttou@Inno_Kids:~$ docker push ayaahmyttou/inno_kids:postgres
The push refers to repository [docker.io/ayaahmyttou/inno_kids]
15bc899d0ab1: Mounted from library/postgres
a2c780dbd0b0: Mounted from library/postgres
e0d67779fb00: Mounted from library/postgres
7ef777167f00: Mounted from library/postgres
7e885641cd61: Mounted from library/postgres
f7e33f1f06ff: Mounted from library/postgres
e3d91581b781: Mounted from library/postgres
c779a0f4ca41: Mounted from library/postgres
2aa7c2a1a6ab: Mounted from library/postgres
1edf901263d3: Mounted from library/postgres
1a74d1d554b0: Mounted from library/postgres
semf3wso17fa: Mounted from library/postgres
1eecha3116fa: Layer already exists
postgres: digest: sha256:37ca99422e2d5133ff9c97fc99db7818fc8b67ea1d8392d86cfb5bb261a0a8 size: 3248
ayaahmyttou@Inno_Kids:~$ 

```

Voici le répertoire Docker Hub pour notre projet "**Inno_Kids**". Il contient trois tags pour différentes images Docker: 'backend', 'postgres', et 'frontinx', indiquant divers composants du projet.

General Tags Builds Collaborators Webhooks Settings

ayaahmyttou/inno_kids

Updated 38 minutes ago

This repository does not have a description   INCOMPLETE

This repository does not have a category   INCOMPLETE

Tags

This repository contains 5 tag(s).

Tag	OS	Type	Pulled	Pushed
 backend		Image	---	38 minutes ago
 postgres		Image	---	4 days ago
 frontinx		Image	---	4 days ago

F-9.Automatisation avec GitHub :

À la conclusion de notre projet, nous avons mis en place une automatisation entre GitHub et Docker Hub. Grâce à l'utilisation de GitHub Actions, chaque modification du code poussée sur GitHub entraîne une mise à jour automatique de l'image correspondante sur Docker Hub. Ce processus est détaillé dans la section dédiée à l'intégration continue et au déploiement continu (CI/CD) de notre documentation. Cette automatisation assure que les dernières versions de nos applications sont toujours disponibles pour le déploiement et l'utilisation, maximisant ainsi l'efficacité et la réactivité de notre flux de travail de développement.

