

Hébergement de site web

Table des matières

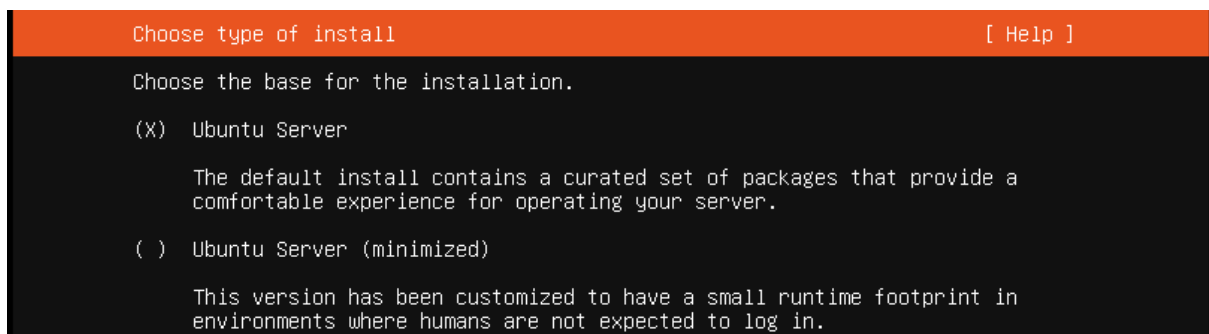
| | |
|----------------------------------------------------------------------------------------|----|
| I) Contextualisation..... | 2 |
| II) Installation du système d'exploitation sur le serveur..... | 3 |
| Première connexion SSH..... | 9 |
| III) Installation du serveur web sur le Ubuntu server..... | 10 |
| IV) Configuration DNS..... | 11 |
| Configuration d'un serveur DNS pour héberger plusieurs sites web sur Ubuntu Linux..... | 11 |
| Étape 1 : Installation du serveur DNS Bind9..... | 12 |
| Étape 2 : Configuration des fichiers de zone..... | 12 |
| Étape 3 : Configuration du fichier « named.conf.local »..... | 13 |
| Étape 4 : Redémarrage du serveur DNS « Bind9 »..... | 13 |
| Étape 5 : Configuration des enregistrements DNS pour chaque domaine..... | 14 |
| Étape 6 : changement de l'adresse IP du « NAMESERVER »..... | 15 |
| Étape 7 : Vérification de la configuration DNS..... | 15 |
| Conclusion :..... | 16 |
| V) Installation et configuration de MySQL..... | 16 |
| Étape 1 : Mise à jour du système..... | 16 |
| Étape 2 : Installation du serveur MySQL..... | 16 |
| Étape 3 : Configuration de MySQL..... | 17 |
| Étape 4 : Vérification du serveur MySQL..... | 17 |
| Étape 5 : Connexion à MySQL..... | 17 |
| VI) Installation et configuration de phpmyadmin..... | 18 |
| Étape 1 : Installation de phpMyAdmin..... | 18 |
| Étape 2 : Configuration de phpMyAdmin..... | 18 |
| Étape 3 : Redémarrage du serveur web..... | 19 |
| Étape 4 : Accès à phpMyAdmin..... | 19 |
| VII) Conclusion..... | 20 |

I) Contextualisation.

- Nous travaillons dans une entreprise d'infrastructure réseau, notre équipe doit créer un serveur capable d'héberger le projet d'une équipe de développeur. Le serveur sera équipé d'un système d'exploitation Ubuntu et sera configuré avec les outils nécessaires pour héberger des sites web. Une fois l'installation et la configuration du système d'exploitation terminé, Il nous faudra configurer une connexion SSH. Cette connexion SSH se fera via l'outil VMRC pour une sécurité optimale. La connexion SSH permettra à chaque membre de l'équipe d'accéder au serveur pour travailler sur le projet à distance de manière efficace.
- Le serveur une fois opérationnelle aura une partie web avec Apache pour héberger les sites web de l'équipe. Il possèdera également un service DNS pour permettre aux sites web d'être accessibles via des noms de domaine. En outre, le serveur doit comporter le service FTP pour permettre le transfert de fichiers.
- Nous nous occuperons également de la création une base de données pour stocker les données du projet de l'équipe de développeur.

II) Installation du système d'exploitation sur le serveur.

- Pour ce server nous avons choisi Ubuntu comme système d'exploitation. En effet, Ubuntu est un choix populaire pour un serveur pour plusieurs raisons :
 - ⇒ Sa stabilité. En effet Ubuntu est connu pour être stable et fiable, avec des mises à jour régulières pour assurer la sécurité et la stabilité du système.
 - ⇒ Sa facilité d'utilisation : Ubuntu est convivial et facile à utiliser pour les débutants, avec une interface utilisateur graphique accessible et une grande communauté d'utilisateurs qui peuvent aider à résoudre les problèmes.
 - ⇒ Une grande communauté : Ubuntu bénéficie d'une grande communauté de développeurs et d'utilisateurs, ainsi que d'un support professionnel de Canonical, l'entreprise derrière Ubuntu.
 - ⇒ Sa compatibilité : Ubuntu est compatible avec une large gamme de matériels et de logiciels, ce qui en fait un choix idéal pour les serveurs de développement.
 - ⇒ Sa sécurité : Ubuntu jouit d'une réputation solide en matière de sécurité, avec des mises à jour régulières pour protéger contre les vulnérabilités de sécurité.
- Pour l'installation de Ubuntu, nous avons choisi Ubuntu server :



```
Choose type of install [ Help ]

Choose the base for the installation.

(X) Ubuntu Server
    The default install contains a curated set of packages that provide a
    comfortable experience for operating your server.

( ) Ubuntu Server (minimized)
    This version has been customized to have a small runtime footprint in
    environments where humans are not expected to log in.
```

- En effet l'installation du Ubuntu server (minimized aurait engendré une restriction de la quantité de logiciels et de fonctionnalités préinstallé sur le système. En somme Ubuntu Server est la version complète de Ubuntu destinée aux serveurs et inclut une gamme complète de logiciels et de fonctionnalités, tels que, des outils de surveillance du système et de la sécurité.

- Lors de l'installation d'Ubuntu Server, il y a une phase dédiée au paramétrage de la connexion réseau. Cette phase est importante car elle permet à votre serveur de se connecter à Internet ou à votre réseau local. Dans cette phase, il y a possibilité de configurer la connexion réseau de plusieurs manières. Par exemple l'utilisation du DHCP pour obtenir automatiquement une adresse IP, ce pendant il est possible de configurer manuellement une adresse IP statique.
- La capture ci-dessous montre qu'un DHCP a été utilisé pour le paramétrage de la connexion réseau :

```
Connections réseau [ Help ]

Configurez au moins une interface pour que ce serveur puisse communiquer avec
les autres machines sur le réseau, préférentiellement un réseau avec accès aux
mises à jour.

NAME    TYPE    NOTES
[ ens160 eth - ▶ ]
DHCPv4  192.168.40.67/24
00:0c:29:22:af:fa / VMware / VMXNET3 Ethernet Controller

[ Create bond ▶ ]
```

- L'étape suivante consiste maintenant à configurer le miroir d'archive Ubuntu. Un miroir d'archive Ubuntu est un serveur qui contient une copie des packages Ubuntu, qui peuvent être utilisés pour installer et mettre à jour des logiciels sur votre système. En configurant un miroir d'archive, il est possible d'accélérer le processus de téléchargement et d'installation des packages en utilisant un serveur local plutôt que de télécharger à partir d'Internet.
- Comme le démontre la figure ci-dessous, l'adresse du miroir d'archive Ubuntu que nous avons configuré est <http://fr.archive.ubuntu.com/ubuntu>. Cela signifie que nous utilisons un miroir local pour télécharger les packages Ubuntu :

```
Configurer le miroir d'archive Ubuntu [ Help ]

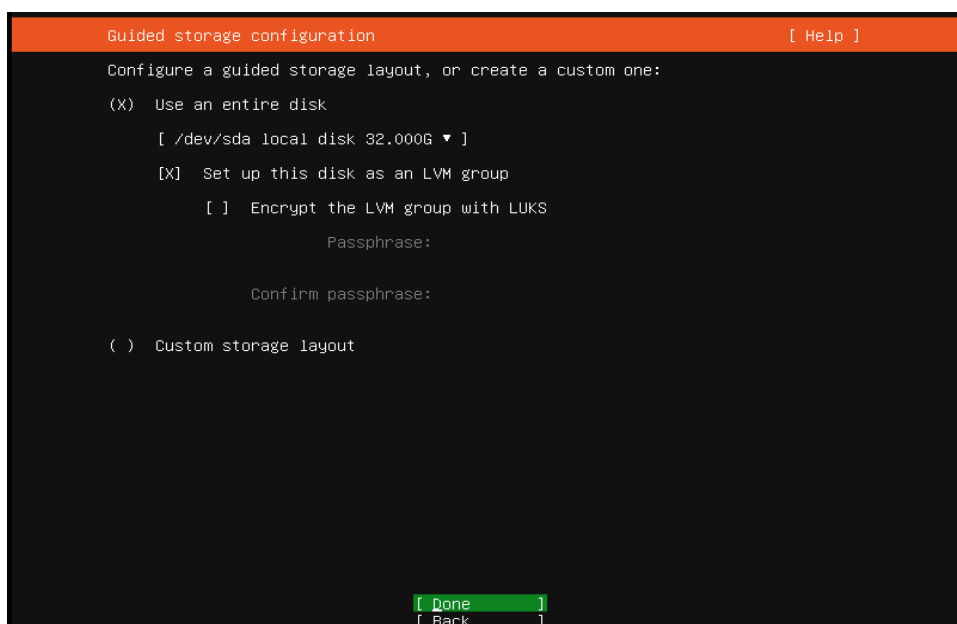
If you use an alternative mirror for Ubuntu, enter its details here.

Adresse du miroir : http://fr.archive.ubuntu.com/ubuntu
You may provide an archive mirror that will be used
instead of the default.
```

- Nous rencontrons ensuite l'étape « guided storage configuration », cette dernière permet de configurer le stockage du système. Plus précisément, cette

étape permet de sélectionner un mode de partitionnement pour le disque dur. On peut voir ici que plusieurs choix s'offrent à nous tel que

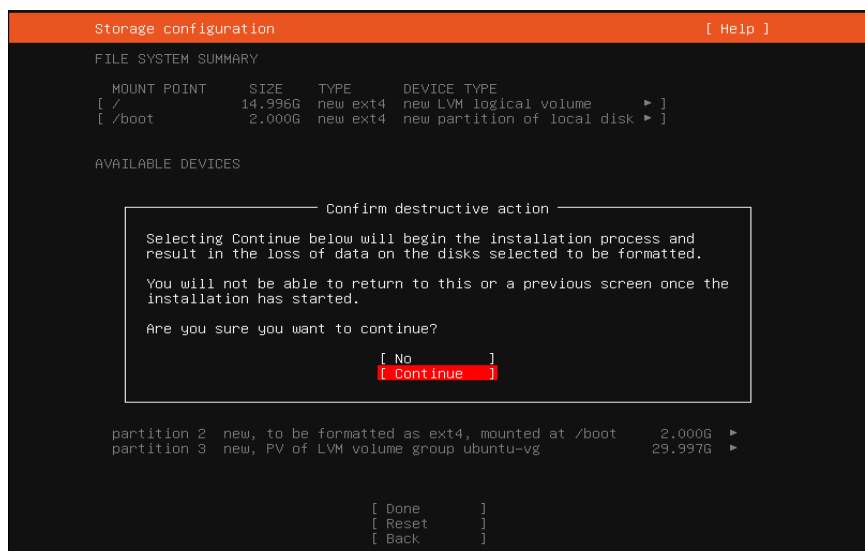
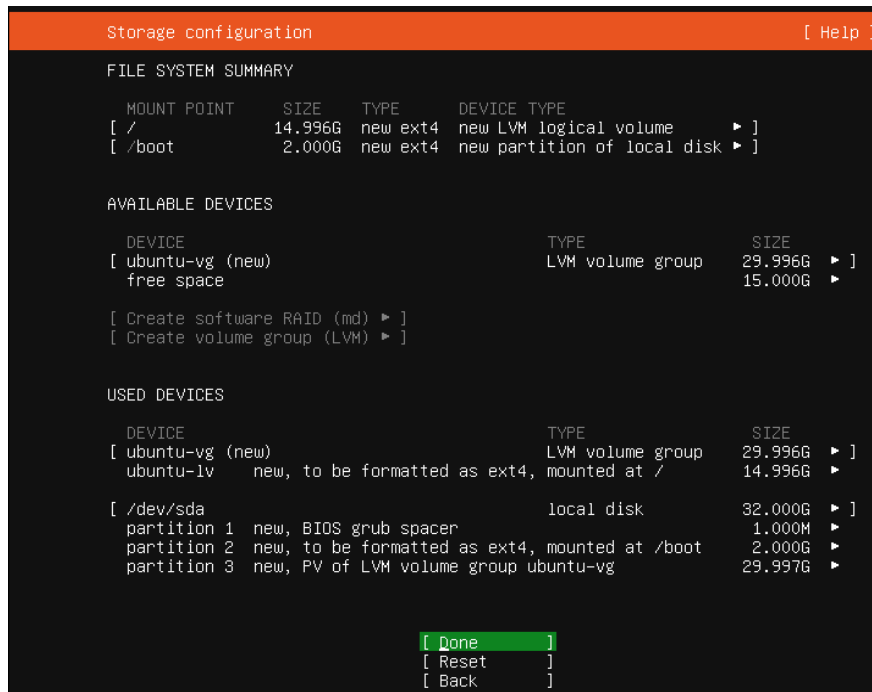
- ⇒ Utiliser tout le disque : Cette option formate tout le disque dur et utilise l'espace disponible pour installer Ubuntu Server et stocker des données.
 - ⇒ Utiliser tout le disque avec LVM : Cette option formate tout le disque dur et utilise l'espace disponible pour installer Ubuntu Server en utilisant le Logical Volume Manager (LVM). LVM est un système de gestion de volumes qui permet de gérer facilement les partitions et de les redimensionner à mesure que les besoins de stockage évoluent.
 - ⇒ Utiliser le disque dur avec des partitions personnalisées : Cette option permet de créer des partitions personnalisées pour le système en utilisant l'outil de partitionnement de disque. Il est possible de spécifier la taille et le type de partition pour chaque partition créée.
- Dans notre cas, la capture ci-dessous montre que le choix sélectionné est d'utiliser tout le disque. Ce choix est une option simple et rapide pour installer Ubuntu Server. Cette option est utile car nous avons besoin d'un système simple et rapide à installer, en revanche on prend soin de noter que celle-ci ne peut pas être la meilleure option si nous avons des besoins de stockage plus complexes. De plus il n'est pas possible d'organiser le système de stockage de manière spécifique :



- L'étape suivante concerne la configuration des partitions de stockage en créant une ou plusieurs partitions avec des tailles et des systèmes de fichiers

spécifiques. Cette option est utile pour personnaliser le stockage du système en fonction de nos besoins spécifiques, mais peut être plus complexe que la configuration guidée.

- Les deux prochaines figures montrent les choix que nous avons appliqués concernant cette phase de la configuration :



- L'une des dernières étapes de la phase de configuration consistait à configurer un profil utilisateur pour notre système comme son nom l'indique (profile setup). Cette étape permet de définir des informations utilisateur, telles que le nom, le nom du serveur, le nom d'utilisateur ou encore le mot de passe.
- La figure ci-dessous montre un exemple des champs rempli :

Profile setup [Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name: PIERRE.BARBIER

Your server's name: puya
The name it uses when it talks to other computers.

Pick a username: pierre

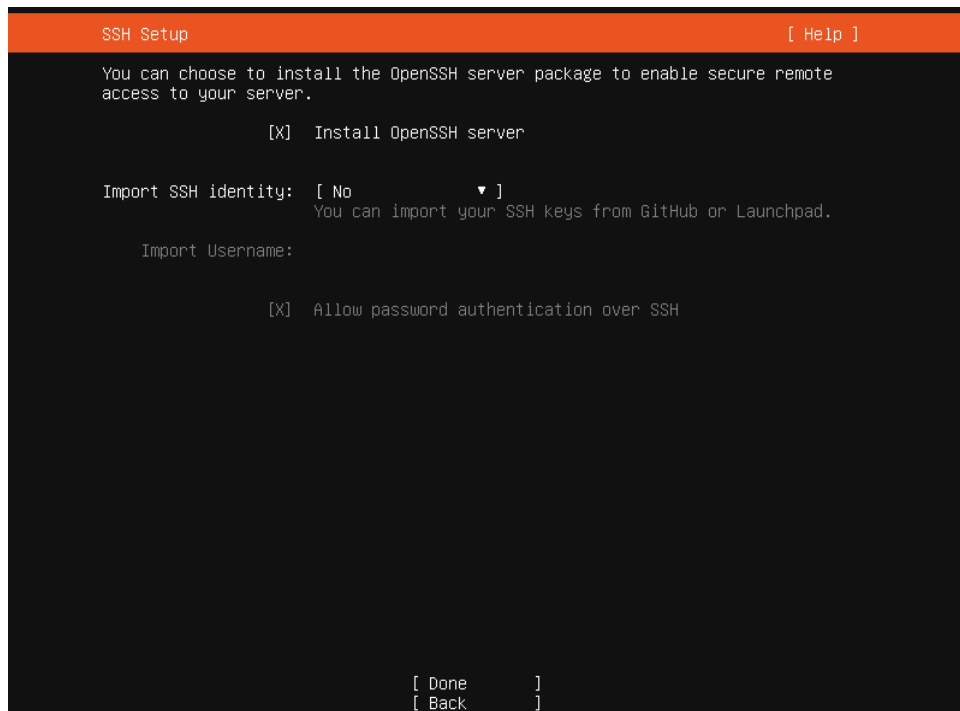
Choose a password: xxxxxxxx

Confirm your password: xxxxxxxx

[Done]

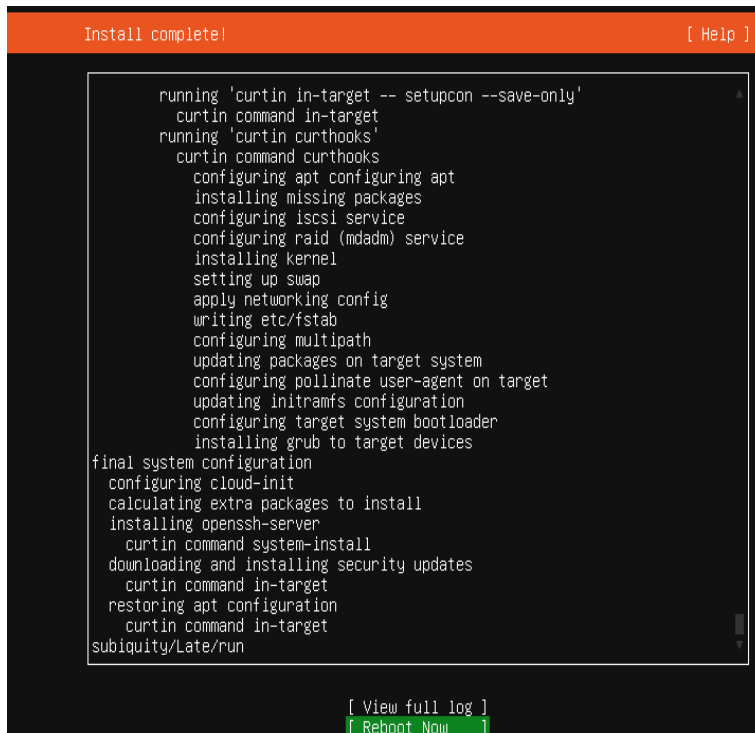
- La phase "Profile Setup" dans Ubuntu Server est une étape optionnelle qui permet de configurer un profil utilisateur pour notre système. Cette étape est utile pour la gestion des utilisateurs et peut être personnalisée pour répondre à des besoins spécifiques.
- La dernière étape concerne la configuration du protocole SSH. Il faut savoir que SHHS est un protocole de communication réseau sécurisé. Cette étape

permet d'accéder au serveur à distance de manière sécurisée. La figure ci-dessous montre que l'option sélectionné est « install open SSH server ». Cette option permet installer le serveur OpenSSH sur votre système Ubuntu, ce qui permet à d'autres ordinateurs de se connecter au serveur via SSH.



- Une fois la configuration du SSH terminé il est possible de se connecter au serveur à distance à l'aide d'un client SSH. Cela permet d'exécuter des commandes sur le serveur à partir de n'importe quel ordinateur connecté à Internet.

- Enfin cette dernière capture montre que l'installation de Ubuntu server c'est déroulé avec succès.



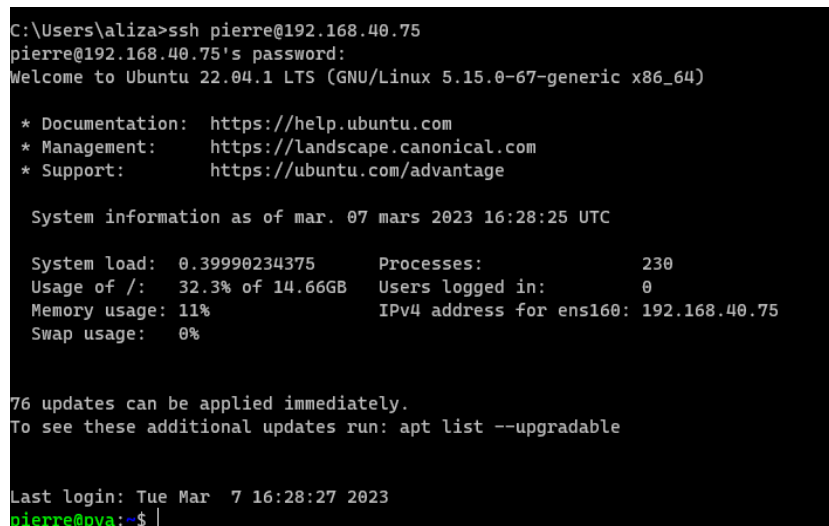
```
running 'curtin in-target -- setupcon --save-only'
curtin command in-target
running 'curtin curthooks'
curtin command curthooks
  configuring apt
  configuring apt
  installing missing packages
  configuring iscsi service
  configuring raid (mdadm) service
  installing kernel
  setting up swap
  apply networking config
  writing etc/fstab
  configuring multipath
  updating packages on target system
  configuring pollinate user-agent on target
  updating initramfs configuration
  configuring target system bootloader
  installing grub to target devices
final system configuration
  configuring cloud-init
  calculating extra packages to install
  installing openssh-server
  curtin command system-install
  downloading and installing security updates
  curtin command in-target
  restoring apt configuration
  curtin command in-target
subiquity/Late/run

[ View full log ]
[ Reboot Now ]
```

- Il faut maintenant redémarrer la machine pour finaliser l'installation.

Première connexion SSH.

- La figure ci-dessous montre que la configuration du SSH est concluante car nous avons réussi à nous connecter au serveur Ubuntu par le biais de SSH.



```
C:\Users\aliza>ssh pierre@192.168.40.75
pierre@192.168.40.75's password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-67-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of mar. 07 mars 2023 16:28:25 UTC

System load:  0.39990234375   Processes:            230
Usage of /:   32.3% of 14.66GB Users logged in:             0
Memory usage: 11%           IPv4 address for ens160: 192.168.40.75
Swap usage:   0%

76 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Tue Mar  7 16:28:27 2023
pierre@pya:~$
```

III) Installation du serveur web sur le Ubuntu server.

- On commence par mettre à jour les paquet déjà présent grâce à la commande « `sudo apt-get update` »
- Ensuite on procède à l'installation grâce à la commande « `sudo apt-get install apache2` ».
- Pour vérifier que l'installation est correctement effectué, on utilise la commande « `sudo systemctl status apache2` » elle permet de vérifier que le service est correctement installé et qu'il tourne correctement sur le serveur.
- La figure ci-dessous montre que l'installation est réussie et que le service est fonctionnel :

```
pierre@pya:/usr$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-03-14 14:27:45 UTC; 9min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 36924 (apache2)
    Tasks: 55 (limit: 2234)
   Memory: 4.8M
      CPU: 91ms
   CGroup: /system.slice/apache2.service
           └─36924 /usr/sbin/apache2 -k start
             └─36926 /usr/sbin/apache2 -k start
               └─36927 /usr/sbin/apache2 -k start
```

- Afin de vérifier qu'apache fonctionne correctement via une interface graphique, on peut effectuer un test en se connectant au serveur. La figure ci-dessous montre que le serveur apache est bien fonctionnel :



- En ce qui concerne le paramétrage d'apache, celui-ci ne nécessite pas de configuration spécifique pour l'utilisation que nous en faisons.

IV) Configuration DNS

Configuration d'un serveur DNS pour héberger plusieurs sites web sur Ubuntu Linux.

- Un serveur DNS (Domain Name System) est un système de noms de domaines qui permet de traduire les noms de domaine en adresses IP. Cette traduction permet aux utilisateurs d'accéder à un site web en tapant simplement son nom de domaine dans leur navigateur web. Dans ce guide, nous allons expliquer comment configurer un serveur DNS pour héberger plusieurs sites web sur un serveur Ubuntu Linux.

Étape 1 : Installation du serveur DNS Bind9

- Le serveur DNS Bind9 est le serveur DNS le plus populaire sous Linux. Vous pouvez l'installer en exécutant les commandes suivantes :
- `sudo apt-get update`
- `sudo apt-get install bind9 bind9utils bind9-doc`

Étape 2 : Configuration des fichiers de zone

- La configuration de Bind9 se fait en éditant les fichiers de zone. Un fichier de zone contient les informations sur un domaine particulier, telles que les adresses IP de ses serveurs, ses enregistrements DNS, etc.
- Créez un fichier de zone pour chaque domaine que vous souhaitez héberger sur votre serveur DNS. Pour cela, créez un fichier de configuration de zone pour chaque domaine dans le répertoire `/etc/bind/zones` :
- `sudo mkdir /etc/bind/zones`
- `cd /etc/bind/zones`
- `sudo nano siteGroupe3.fr.zone`
- Vous pouvez remplacer `example.com` par le nom de domaine que vous souhaitez héberger. Le fichier de configuration de zone ressemble à ceci :

```
GNU nano 6.2 /etc/bind/zones/siteGroupe3.fr.z
$TTL 86400
@      IN      SOA      ns1.siteGroupe3.fr. admin.siteGroupe3.fr. (
                                2016031401      ; Serial
                                3600              ; Refresh
                                1800              ; Retry
                                604800            ; Expire
                                86400            ; Minimum TTL
)

@      IN      NS       ns1.siteGroupe3.fr.
@      IN      NS       ns2.siteGroupe3.fr.
@      IN      MX       10 mail.siteGroupe3.fr.

@      IN      A        192.168.40.75
www    IN      A        192.168.40.75
blog   IN      A        192.168.40.75
```

- Dans cet exemple, nous avons configuré le domaine `siteGroupe3.fr` pour qu'il pointe vers l'adresse IP `192.168.40.75`.

Étape 3 : Configuration du fichier « named.conf.local »

- Le fichier named.conf.local contient les informations de configuration pour chaque zone hébergée sur votre serveur DNS. Éditez ce fichier en exécutant la commande suivante :
 - `sudo nano /etc/bind/named.conf.local`
- Ajoutez les zones que vous avez configurées dans le fichier de configuration de zone en ajoutant les lignes suivantes à la fin du fichier named.conf.local :

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

zone "siteGroupe3.fr" {
    type master;
    file "/etc/bind/zones/siteGroupe3.fr.zone";
};

zone "autre-siteGroupe3.fr" {
    type master;
    file "/etc/bind/zones/autre-siteGroup3.fr.zone";
};
```

- Assurez-vous de remplacer les noms de domaine par ceux que vous avez configurés.

Étape 4 : Redémarrage du serveur DNS « Bind9 »

- Après avoir configuré les fichiers de zone et le fichier named.conf.local, vous devez redémarrer le serveur DNS Bind9 en exécutant la commande suivante :
 - `sudo systemctl restart bind9`
- Votre serveur DNS est maintenant configuré pour héberger plusieurs sites web.

Étape 5 : Configuration des enregistrements DNS pour chaque domaine

- Maintenant que vous avez configuré votre serveur DNS, vous devez configurer les enregistrements DNS pour chaque domaine que vous souhaitez héberger sur votre serveur.
- Les enregistrements DNS sont utilisés pour définir les adresses IP de chaque serveur de votre domaine, ainsi que pour définir les enregistrements de messagerie et les alias de domaine.
- Pour ajouter un enregistrement DNS pour un domaine, éditez le fichier de zone correspondant en exécutant la commande suivante :
 - `sudo nano /etc/bind/zones/siteGroupe3.fr.zone`
- Ajoutez les enregistrements DNS requis pour votre domaine. Par exemple, pour ajouter un enregistrement DNS pour un sous-domaine appelé "blog" qui pointe vers l'adresse IP 192.168.40.75, ajoutez la ligne suivante à votre fichier de zone :
 - `blog IN A 192.168.40.75`
- Assurez-vous de remplacer "example.com" par le nom de domaine que vous avez configuré.
- Vous pouvez également ajouter des enregistrements de messagerie pour votre domaine en ajoutant des enregistrements MX (Mail Exchange) à votre fichier de zone. Par exemple, pour définir l'enregistrement MX pour le domaine example.com, ajoutez la ligne suivante à votre fichier de zone :
 - `@ IN MX 10 mail.siteGroupe3.fr.`
- Assurez-vous de remplacer " siteGroupe3.fr " par le nom de domaine que vous avez configuré et "mail. siteGroupe3.fr " par le nom de domaine de votre serveur de messagerie.

Étape 6 : changement de l'adresse IP du « NAMESERVER »

- Ouvrez le fichier de configuration de resolv.conf en utilisant votre éditeur de texte préféré avec la commande :
 - `sudo nano /etc/resolv.conf`
- Ensuite, il faut trouver la ligne "nameserver" et remplacez l'adresse IP existante par la nouvelle adresse IP du serveur de noms que vous souhaitez utiliser. Si la ligne "nameserver" n'existe pas, vous pouvez l'ajouter en bas du fichier comme suit :
 - Nameserver 192.168.40.75
- Enregistrez et fermez le fichier en appuyant sur Ctrl + X, puis en appuyant sur Y pour confirmer les modifications et enfin sur Entrée pour enregistrer le fichier.
- Redémarrez le service réseau pour prendre en compte les modifications :
 - `sudo systemctl restart networking.service`
- Votre serveur Ubuntu devrait maintenant utiliser la nouvelle adresse IP pour le serveur de noms.

Étape 7 : Vérification de la configuration DNS

- Après avoir configuré les enregistrements DNS pour chaque domaine, vous pouvez vérifier la configuration de votre serveur DNS en utilisant l'outil dig. Par exemple, pour vérifier les enregistrements DNS pour le domaine example.com, exécutez la commande suivante :
 - `dig siteGroupe3.fr`
- Cela vous donnera les informations DNS pour le domaine example.com, telles que les adresses IP des serveurs de votre domaine, les enregistrements de messagerie, etc.

Conclusion :

- La configuration d'un serveur DNS pour héberger plusieurs sites web sur Ubuntu Linux est un processus relativement simple qui peut être réalisé en suivant les étapes décrites dans ce guide. Assurez-vous de configurer les fichiers de zone pour chaque domaine que vous souhaitez héberger, de configurer les enregistrements DNS pour chaque domaine, et de vérifier la configuration DNS en utilisant l'outil dig.
- Cependant, il est important de noter que la configuration DNS peut varier en fonction de vos besoins spécifiques. Par exemple, si vous utilisez un fournisseur de services DNS tiers pour gérer les enregistrements DNS de votre domaine, vous devrez peut-être configurer des enregistrements de redirection de votre fournisseur de services DNS vers votre serveur DNS.

```
root@pya:/# dig siteGroupe3.fr
; <<>> DiG 9.18.1-1ubuntu1.3-Ubuntu <<>> siteGroupe3.fr
; global options: +cmd
; Got answer:
; ->>HEADER<- opcode: QUERY, status: NXDOMAIN, id: 61198
; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
; QUESTION SECTION:
;siteGroupe3.fr.
; IN A
; AUTHORITY SECTION:
fr. 600 IN SOA a.nic.fr. dnsmaster.afnic.fr. 2235106851 3600 1800 36000000 600
; Query time: 60 msec
; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
; WHEN: Tue Mar 14 15:29:08 UTC 2023
; MSG SIZE rcvd: 101
```

V) Installation et configuration de MySQL

Étape 1 : Mise à jour du système

Vérifier si tous les paquets sont à jour en exécutant les commandes suivantes :

```
sudo apt-get update
sudo apt-get upgrade
```

Étape 2 : Installation du serveur MySQL

Commencez par installer le paquets mysql-server en faisant la commande suivante :

```
sudo apt-get install mysql-server
```

De ce fait l'installation du paquet s'effectue et la commande lancera une série de scripts de configuration, notamment la définition du mot de passe root pour MySQL.

Étape 3 : Configuration de MySQL

À la fin de l'installation de MySQL, vous pouvez exécuter la commande suivante pour sécuriser l'installation :

```
sudo mysql_secure_installation
```

Vous serez guidée tout au long du processus de sécurisation de votre installation MySQL, en vous permettant de modifier le mot de passe root et de supprimer les utilisateurs anonymes, les bases de données de test et les connexions distantes à MySQL.

Étape 4 : Vérification du serveur MySQL

Pour vous assurer que le serveur MySQL est en cours d'exécution, exécutez la commande suivante :

```
systemctl status mysql.service
```

Si MySQL est en cours d'exécution, alors vous verrez une sortie similaire à celle-ci :

```
pierre@pya:~$ systemctl status mysql.service
● mariadb.service - MariaDB 10.6.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-04-04 13:33:40 UTC; 2 weeks 6 days ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 31972 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 7 (limit: 2227)
    Memory: 17.1M
         CPU: 9min 54.623s
    CGroup: /system.slice/mariadb.service
            └─31972 /usr/sbin/mariabdd
```

Étape 5 : Connexion à MySQL

Vous pouvez vous connecter à votre MySQL en utilisant la commande suivante :

```
mysql -u root -p
```

Ce qui vous invitera à entrer le mot de passe root MySQL, entrez le mot de passe que vous avez défini à l'étape 3, et vous pourriez interagir directement avec votre MySQL.

VI) Installation et configuration de phpmyadmin

Étape 1 : Installation de phpMyAdmin

Pour installer phpMyAdmin, nous partons du principe que vous avez vérifié si votre système et ses paquets sont à jour. Si ce n'est pas le cas, je vous invite à aller voir les commandes qui sont présentes dans l'installation et la configuration de MySQL. Si tout est à jour, vous pouvez exécuter la commande suivante :

```
sudo apt-get install phpmyadmin
```

L'installation de phpMyAdmin débutera et lancera une série de scripts de configuration, notamment la définition du mot de passe pour le compte administrateur de phpMyAdmin. Au cours de l'installation, vous pouvez être invité à choisir un serveur web à utiliser. Si vous utilisez Apache, sélectionnez Apache. Si vous utilisez Nginx ou un autre serveur web, ne sélectionnez aucun.

Lorsque vous êtes invité à configurer la base de données pour phpMyAdmin, sélectionnez Oui et entrez le mot de passe root MySQL.

Étape 2 : Configuration de phpMyAdmin

Une fois l'installation terminée, vous devez configurer phpMyAdmin pour qu'il fonctionne avec votre installation MySQL. Ouvrez le fichier de configuration de phpMyAdmin en exécutant la commande suivante :

```
sudo nano /etc/dbconfig-common/phpmyadmin.conf
```

Trouvez la ligne qui commence par "dbc_dbuser =" et remplacez la valeur par "phpmyadmin".

```
# dbc_dbuser: database user
#         the name of the user who we will use to connect to the database.
dbc_dbuser='phpmyadmin'
```

Trouvez la ligne qui commence par "dbc_dbpass =" et remplacez la valeur par le mot de passe de l'utilisateur phpmyadmin.

Enregistrez et fermez le fichier.

Étape 3 : Redémarrage du serveur web

Après avoir configuré phpMyAdmin, vous devez redémarrer votre serveur web pour qu'il puisse prendre en compte les modifications et ce synchronisez avec la base de données et phpmyadmin.

Si vous utilisez Apache, exécutez la commande suivante :

```
sudo service apache2 restart
```

Si vous utilisez Nginx ou un autre serveur web, exécutez la commande appropriée pour redémarrer le serveur web.

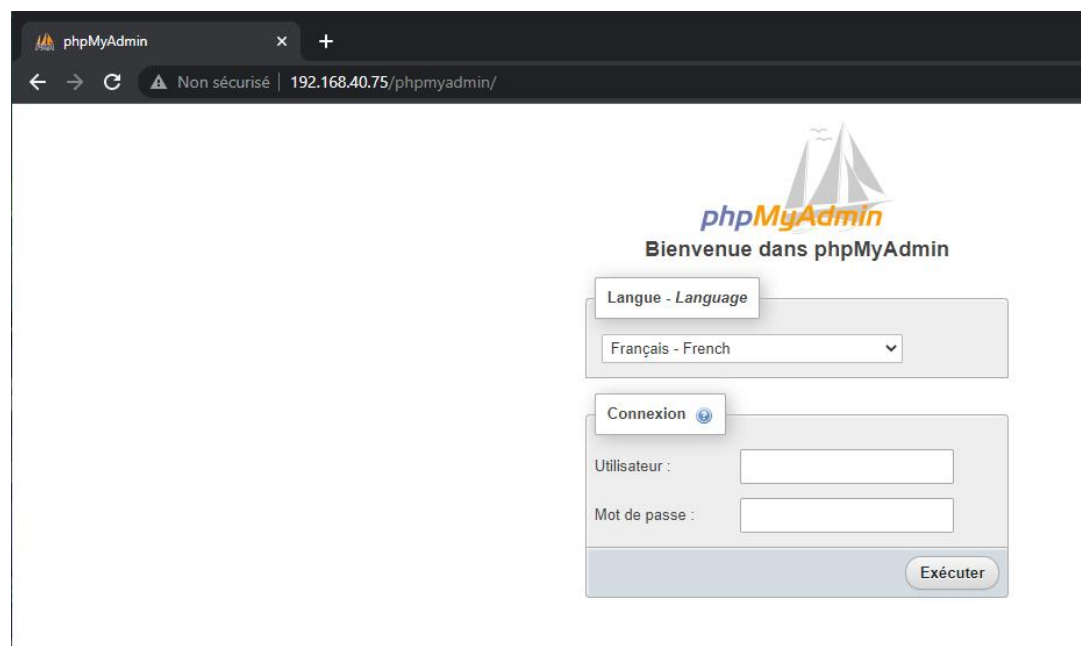
Étape 4 : Accès à phpMyAdmin

Pour vérifier si phpMyAdmin fonctionne ouvrez votre navigateur Web et accédez à l'adresse suivante :

```
http://votre-serveur-ip/phpmyadmin/
```

Remplacez "votre-serveur-ip" par l'adresse IP de votre serveur.

Une fois cela fait vous devriez voir l'interface de connexion de phpMyAdmin. Entrez le nom d'utilisateur et le mot de passe que vous avez définis lors de l'installation de phpMyAdmin, et voilà vous pouvez utiliser l'interface pour gérer vos bases de données MySQL.



VII) Conclusion

Enfin, l'équipe d'infrastructure réseau s'est chargée de créer un serveur robuste et sécurisé pour héberger les projets de l'équipe de développement. Grâce à l'utilisation du système d'exploitation Ubuntu et à une configuration soigneusement sélectionnée, les serveurs sont prêts à héberger votre site Web avec une grande efficacité. L'établissement d'une connexion SSH via les outils VMRC garantit une sécurité optimale, permettant à tous les membres de l'équipe d'accéder à distance au serveur et de collaborer sur des projets.

Outre l'hébergement de sites Web, le serveur dispose également de fonctions importantes telles qu'Apache pour la gestion des pages Web, le service DNS pour accéder aux sites Web via des noms de domaine et le service FTP pour les transferts de fichiers. De plus, nous construisons une base de données pour stocker les données du projet et réaliser une gestion efficace et sûre des informations. Grâce à notre expertise en infrastructure réseau et à notre engagement à fournir des solutions de qualité, nous sommes convaincus que ce serveur répondra aux besoins de votre équipe de développement. En fournissant une plate-forme robuste et sécurisée pour le développement et l'hébergement, nous contribuons au succès du projet tout en offrant une expérience de travail à distance efficace et collaborative.