

tape 1 — Installation d'Apache

Apache est disponible dans les référentiels de logiciels par défaut de Linux, ce qui permet de l'installer à l'aide d'outils classiques de gestion des packages.

Commençons par mettre à jour l'index local des packages pour refléter tout nouveau changement en amont :

```
$ sudo apt update
```

Ensuite, installez le package `apache2` :

```
$ sudo apt install apache2
```

Après avoir confirmé l'installation, `apt` installera Apache et toutes les dépendances requises.

Il faut démarrer le serveur `apache2`

```
$ /etc/init.d/apache2 start
```

```
(kali@kali)-[~]  
$ /etc/init.d/apache2 start  
Starting apache2 (via systemctl): apache2.service.
```

Étape 2 — Réglage du pare-feu

Avant de tester Apache, il est nécessaire de modifier les paramètres du pare-feu pour permettre à l'extérieur d'accéder aux ports web par défaut. En supposant que vous ayez suivi les instructions des conditions préalables, vous devriez avoir un pare-feu UFW configuré pour restreindre l'accès à votre serveur.

Lors de l'installation, Apache s'enregistre auprès de l'UFW pour fournir quelques profils d'application qui peuvent être utilisés pour activer ou désactiver l'accès à Apache à travers le pare-feu.

Listez les profils d'application `ufw` en tapant :

```
$ sudo ufw app list
```

Si n'est pas installé il faut l'installer :

```
$ sudo ufw app ufw OU
```

```
$ ufw  
Command 'ufw' not found, but can be installed with:  
sudo apt install ufw  
Do you want to install it? (N/y)y  
sudo apt install ufw  
Installing:  
ufw  
Suggested packages:  
rsyslog
```

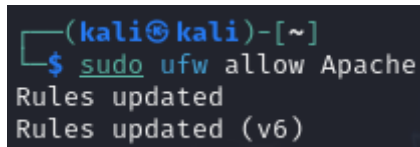
```
(kali@kali)-[~]  
$ sudo ufw app list  
Available applications:  
AIM  
Apache  
Apache Full  
Apache Secure  
Bonjour  
CIFS  
DNS  
Deluge  
IMAP  
IMAPS  
IPP  
KTorrent  
Kerberos Admin  
Kerberos Full  
Kerberos KDC  
Kerberos Password  
LDAP  
LDAPS  
LPD  
MSN  
MSN SSL  
Mail submission  
NFS  
Nginx Full  
Nginx HTTP  
Nginx HTTPS  
Nginx QUIC  
OpenSSH  
POP3  
POP3S  
PeopleNearby  
SMTP  
SSH  
Samba  
Socks  
Telnet  
Transmission  
Transparent Proxy
```

Comme le montre la sortie, il existe trois profils pour Apache :

- **Apache** : ce profil n'ouvre que le port 80 (trafic web normal, non crypté)
- **Apache Full** : ce profil ouvre à la fois le port 80 (trafic web normal, non port 443 (trafic crypté TLS/SSL).
- **Apache Secure** : ce profil n'ouvre que le port 443 (trafic crypté TLS/SS

Nous vous recommandons d'activer le profil le plus restrictif qui autorisera tout de même le trafic que vous avez configuré. Comme nous n'avons pas encore configuré le SSL pour notre serveur dans ce guide, nous devons seulement autoriser le trafic sur le port 80 :

```
$ sudo ufw allow 'Apache'
```



```
(kali㉿kali)-[~]  
$ sudo ufw allow Apache  
Rules updated  
Rules updated (v6)
```

Vous pouvez vérifier le changement en saisissant :

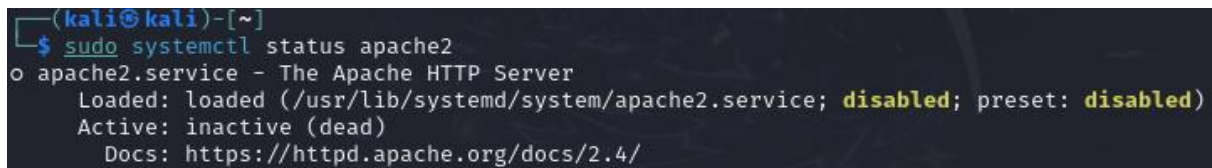
```
$ sudo ufw status
```

Étape 3 - Vérification de votre serveur Web

À la fin du processus d'installation, Kali lance Apache. Le serveur Web devrait déjà être opérationnel.

Vérifiez avec le système `systemd` init pour vous assurer que le service fonctionne en tapant :

```
$ sudo systemctl status apache2
```




```
(kali㉿kali)-[~]  
$ sudo systemctl status apache2  
o apache2.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: disabled)  
   Active: inactive (dead)  
   Docs: https://httpd.apache.org/docs/2.4/
```

Comme cette sortie le confirme, le service a démarré avec succès. Cependant, la meilleure façon de le tester consiste à demander une page à Apache.

Vous pouvez accéder à la page d'accueil par défaut d'Apache pour confirmer que le logiciel fonctionne correctement grâce à votre adresse IP : Si vous ne connaissez pas l'adresse IP de votre serveur, vous pouvez l'obtenir de plusieurs façons différentes à partir de la ligne de commande.

Essayez de taper ceci à l'invite de commande de votre serveur :

```
$ hostname -I
```



```
(kali㉿kali)-[~]  
$ hostname -I  
192.168.0.155
```

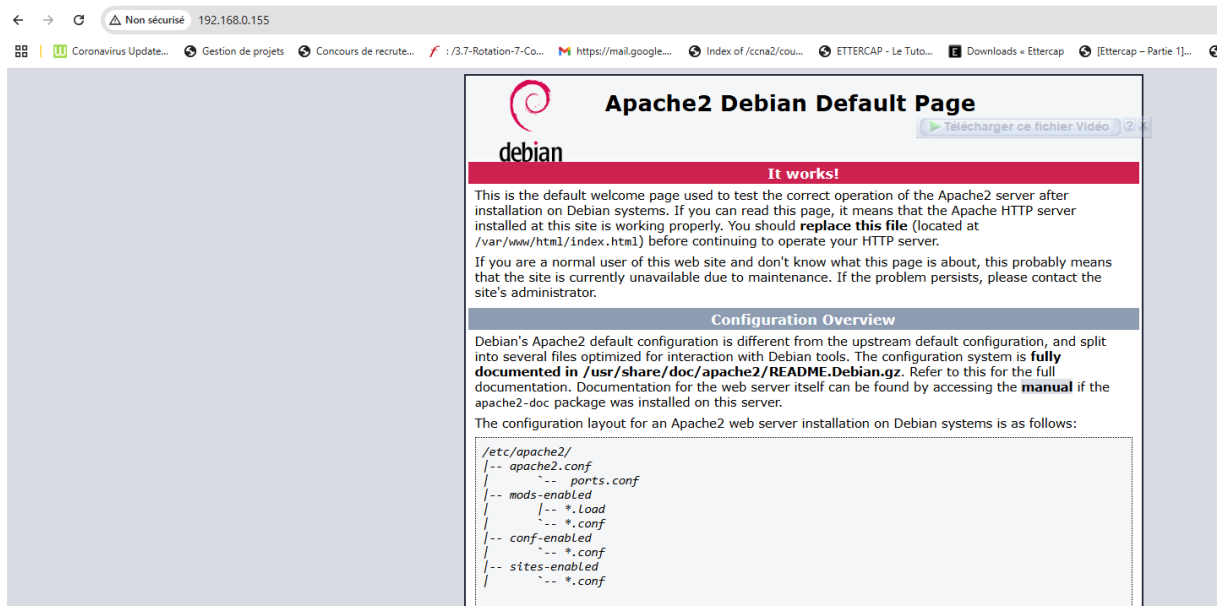
Vous obtiendrez quelques adresses séparées par des espaces. Vous pouvez essayer chacune d'elles dans votre navigateur web pour déterminer s'ils fonctionnent.

Utiliser l'outil `icanhazip`, qui devrait vous donner votre adresse IP publique telle qu'elle est lue depuis un autre endroit sur Internet, constitue une autre option :

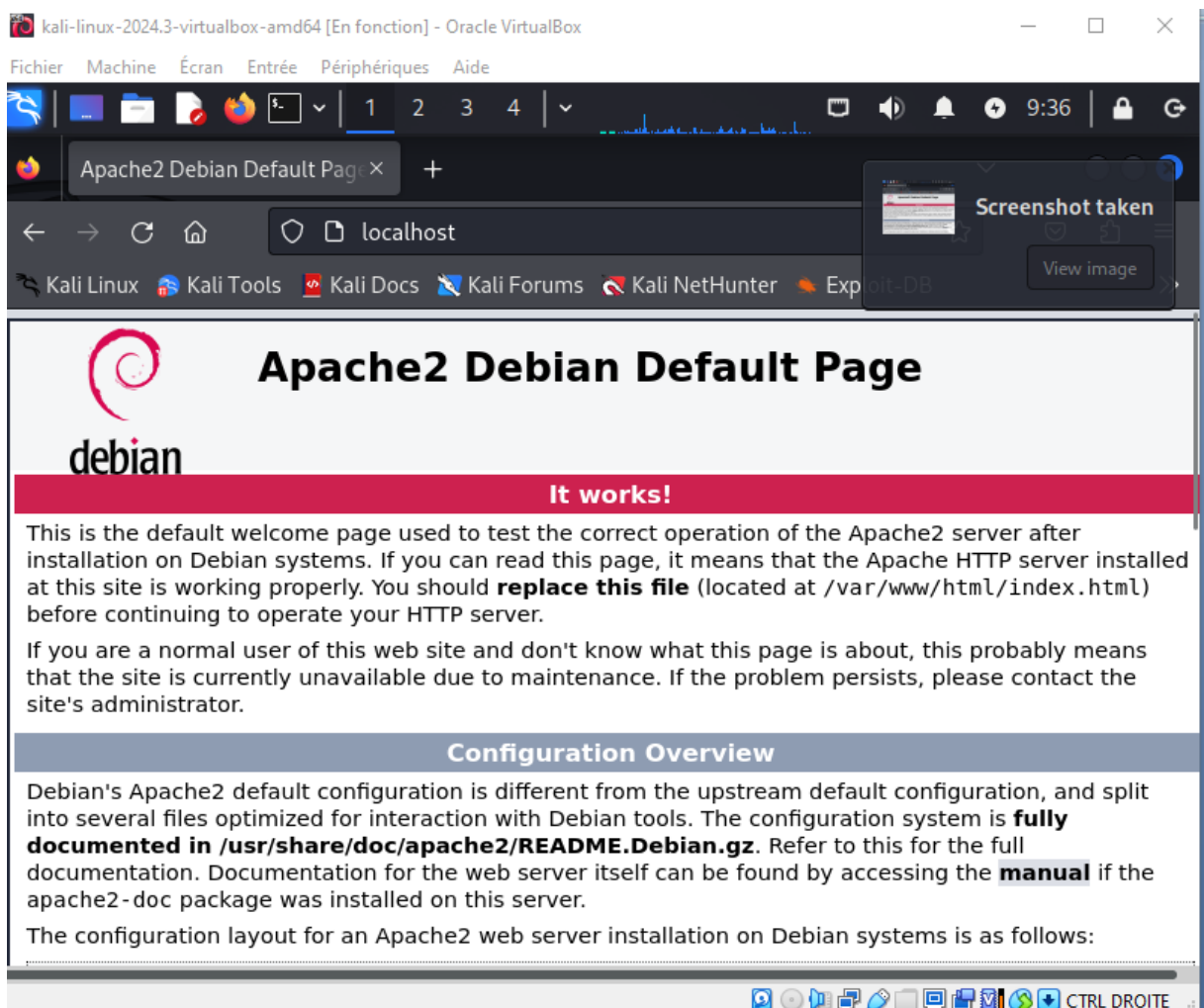
```
$ curl -4 icanhazip.com
```

Une fois que vous avez l'adresse IP de votre serveur, saisissez-la dans la barre d'adresse de votre navigateur : (machine physique sous Windows)

<http://192.168.0.155>



(Machine virtuelle)



Étape 4 — Gestion du processus Apache

Maintenant que votre serveur web est opérationnel, passons en revue quelques commandes de gestion de base en utilisant `systemctl`.

Pour arrêter votre serveur Web, tapez :

```
sudo systemctl stop apache2
```

Pour démarrer le serveur web lorsqu'il est arrêté, tapez :

```
sudo systemctl start apache2
```

Pour arrêter puis redémarrer le service, tapez :

```
sudo systemctl restart apache2
```

Si vous procédez uniquement à des modifications de configuration, il se peut qu'Apache recharge souvent sans interrompre les connexions. Pour ce faire, utilisez cette commande :

```
sudo systemctl reload apache2
```

Par défaut, Apache est configuré pour un lancement automatique au démarrage du serveur. Si ce n'est pas ce que vous souhaitez, désactivez ce comportement en tapant :

```
sudo systemctl disable apache2
```

Pour réactiver le service de lancement automatique au démarrage, tapez :

```
sudo systemctl enable apache2
```

Désormais, Apache devrait démarrer automatiquement au redémarrage du serveur.

Étape 5 — Configuration des hôtes virtuels (recommandé)

Lorsque vous utilisez le serveur web Apache, vous pouvez utiliser des *hôtes virtuels* (similaires aux blocs de serveurs dans Nginx) pour encapsuler les détails de la configuration et héberger plusieurs domaines à partir d'un seul serveur. Nous allons configurer un domaine appelé **your_domain**, mais vous devez **le remplacer par le nom de votre domaine**. (isetzg exemple)

Apache sur KALI a un bloc serveur activé par défaut qui est configuré pour servir des documents à partir du répertoire `/var/www/html`. Même si cela fonctionne bien pour un seul site, cela peut devenir ingérable si vous hébergez plusieurs sites. Au lieu de modifier `/var/www/html`, créons une structure de répertoire au sein de `/var/www` pour un **isetzg**, en laissant `/var/www/html` en place comme répertoire par défaut qui sera utilisé si une demande du client ne correspond à aucun autre site. Créez le répertoire pour **isetzg** comme suit :

Créez le répertoire pour **your_domain (isetzg)** comme suit :

```
sudo mkdir /var/www/iseteg
```

Ensuite, attribuez la propriété du répertoire avec la variable d'environnement `$USER`

```
sudo chown -R $USER:$USER /var/www/isetzg
```

Copy

Les autorisations de la racine de votre site Internet devraient être correctes si vous n'avez pas modifié votre valeur `umask`, qui définit les autorisations de fichier par défaut. Pour vous assurer que vos autorisations sont correctes et permettre au propriétaire de lire, écrire et exécuter les fichiers tout en accordant uniquement des autorisations de lecture et d'exécution aux groupes et aux autres, vous pouvez saisir la commande suivante :

```
sudo chmod -R 755 /var/www/isetzg
```

Ensuite, créez une page `index.html` à l'aide de `nano` ou de votre éditeur favori :

```
sudo nano /var/www/isetzg/index.html
```

À l'intérieur, ajoutez l'exemple de HTML suivant :

```
/var/www/iseteg/index.html
```

```
<html>
  <head>
    <title>BIENVENUE !</title>
  </head>
  <body>
    <h1>bravo!  Bienvenue dans notre institut !</h1>
  </body>
</html>
```

Enregistrez et fermez le fichier lorsque vous avez terminé.

Pour qu'Apache puisse servir ce contenu, il est nécessaire de créer un fichier d'hôte virtuel avec les directives correctes. Au lieu de modifier directement le fichier de configuration par défaut situé dans `/etc/apache2/sites-available/000-default.conf`, créons-en un nouveau dans `/etc/apache2/sites-available/your_domain.conf`:

```
sudo nano /etc/apache2/sites-available/isetzg.conf
```

Collez le bloc de configuration suivant, qui est similaire à celui par défaut, mais mis à jour pour notre nouveau répertoire et notre nom de domaine :

```
/etc/apache2/sites-available/isetzg.conf
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName isetzg
  ServerAlias www.isetzg
  DocumentRoot /var/www/isetzg
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Notez que nous avons mis à jour le `DocumentRoot` dans notre nouveau répertoire et le `ServerAdmin` dans un e-mail auquel l'administrateur du site **isetzg** peut accéder. Nous avons également ajouté deux directives : `ServerName`, qui établit le domaine de base qui doit correspondre à cette définition d'hôte virtuel, et `ServerAlias`, qui définit d'autres noms qui doivent correspondre comme s'ils étaient le nom de base.

Enregistrez et fermez le fichier lorsque vous avez terminé.

Activons le fichier avec l'outil `a2ensite` :

```
sudo a2ensite isetzg.conf
```

Désactivez le site par défaut défini dans `000-default.conf` :

```
sudo a2dissite 000-default.conf
```

Ensuite, effectuons un test à la recherche d'éventuelles erreurs de configuration :

```
sudo apache2ctl configtest
```

Vous devriez voir la sortie suivante :

```
Output
```

```
Syntax OK
```

Redémarrez Apache pour implémenter vos modifications :

```
sudo systemctl restart apache2
```

Apache devrait maintenant vous présenter votre nom de domaine. Vous pouvez le vérifier en allant sur `http://isetzg` où vous devriez voir quelque chose similaire à ceci :

```
bravo!  Bienvenue dans notre institut ! institut !
```

Étape 6 – Familiarisation avec les fichiers et répertoires Apache importants

Maintenant que vous savez comment gérer le service Apache en lui-même, vous devriez prendre quelques minutes pour vous familiariser avec quelques répertoires et fichiers importants.

Contenu

- `/var/www/html` : le contenu web réel (qui par défaut ne consiste qu'en la page Apache par défaut que vous avez vue précédemment), est présenté à partir du répertoire `/var/www/html`. Cela peut être modifié en changeant les fichiers de configuration Apache.

Configuration du serveur

- `/etc/apache2` : le répertoire de configuration Apache. Tous les fichiers de configuration Apache se trouvent ici.
- `/etc/apache2/apache2.conf` : le fichier de configuration principal d'Apache. Celui-ci peut être modifié pour apporter des changements à la configuration globale de Apache. Ce fichier est responsable du chargement de nombreux autres fichiers dans le répertoire de configuration.
- `/etc/apache2/ports.conf` : ce fichier spécifie les ports sur lesquels Apache écoutera. Par défaut, Apache écoute sur le port 80 et sur le port 443 en plus lorsqu'un module fournissant des capacités SSL est activé.
- `/etc/apache2/sites-available/` : le répertoire dans lequel les hôtes virtuels par site peuvent être stockés. Apache n'utilisera pas les fichiers de configuration trouvés dans ce répertoire à moins qu'ils ne soient liés au répertoire `sites-enabled`. En règle générale, la configuration de tous les blocs du serveur se fait dans ce répertoire, puis s'active en établissant une liaison avec l'autre répertoire grâce à la commande `a2ensite`.
- `/etc/apache2/sites-enabled/` : le répertoire dans lequel les hôtes virtuels par site sont stockés. En règle générale, ils sont créés en reliant les fichiers de configuration trouvés dans le répertoire `sites-available` avec `a2ensite`. Apache lit les fichiers de configuration et les liens trouvés dans ce répertoire lorsqu'il démarre ou se recharge pour compiler une configuration complète.
- `/etc/apache2/conf-available/`, `/etc/apache2/conf-enabled/` : ces répertoires ont la même relation que les répertoires `sites-available` et `sites-enabled` mais sont utilisés pour stocker des fragments de configuration qui n'appartiennent pas à un hôte virtuel. Les fichiers dans le répertoire `conf-available` peuvent être activés avec la commande `a2enconf` et désactivés avec la commande `a2disconf`.
- `/etc/apache2/mods-available/`, `/etc/apache2/mods-enabled/` : ces répertoires contiennent respectivement les modules disponibles et activés. Les fichiers se terminant par `.load` contiennent des fragments pour charger des modules spécifiques, tandis que les fichiers se terminant par `.conf` contiennent la configuration de ces modules. Les modules peuvent être activés et désactivés à l'aide des commandes `a2enmod` et `a2dismod`.

Journaux du serveur

- `/var/log/apache2/access.log` : par défaut, chaque demande envoyée à votre serveur web est enregistrée dans ce fichier de journal à moins qu'Apache ne soit configuré pour faire autrement.
- `/var/log/apache2/error.log` : par défaut, toutes les erreurs sont enregistrées dans ce fichier. La directive `LogLevel` dans la configuration Apache précise le nombre de détails que les journaux d'erreur contiendront.