Exercice 2 : HTTPS — Récapitulatif complet étape par étape

Préparation

- 5 machines virtuelles Debian :
  - **CA-Server** (autorité de certification) Console (mode texte)
  - **HTTPS-Server1** (serveur web HTTPS) Console
  - **HTTPS-Server2** (serveur web HTTPS) Console
  - **Client1** (client) Graphique (pour navigateur)
  - **Client2** (client) Graphique (pour navigateur)
- PC hôte Windows sert à faire tourner VMware et gérer les VM.

Étape 1 : Vérification réseau et noms/IP

- Attribue une IP fixe à chaque VM (ou utilise DHCP mais vérifie l'IP).
- Sur chaque machine, note son **nom** (ex: hostname) et son **IP** (ip a ou ifconfig).
- Vérifie la communication réseau entre toutes les machines avec la commande ping :
  - Depuis chaque client vers chaque serveur et le CA-Server
  - Depuis chaque serveur vers les autres serveurs et clients

---

Étape 2 : Création de l'Autorité de Certification (CA) sur CA-Server

- Installe openssl sur CA-Server si ce n'est pas déjà fait.
- Génére une paire de clés privée + certificat auto-signé pour ta CA (certificat racine).
- Garde précieusement la clé privée et le certificat racine (fichier .crt).

Étape 3 : Configuration de la reconnaissance des certificats par tous

- Copie le certificat racine CA (ca.crt) sur :
  - Les 2 serveurs HTTPS
  - Les 2 clients
- Installe ce certificat racine dans le magasin de certificats de confiance sur chaque machine (pour Debian, souvent dans /usr/local/share/ca-certificates/ puis update-ca-certificates).

Étape 4 : Création de certificats HTTPS pour les serveurs

- Sur **CA-Server**, crée des certificats signés par ta CA pour :
  - HTTPS-Server1
  - HTTPS-Server2
- Pour chaque serveur, génère une clé privée et une demande de signature (CSR).
- Signe les CSR avec ta CA pour générer des certificats signés.
- Transfère la clé privée et le certificat signé vers chaque serveur correspondant.

Étape 5 : Configuration des serveurs HTTPS

- Sur HTTPS-Server1 et HTTPS-Server2 :
  - Installe un serveur web léger (exemple : apache2 ou nginx).
  - Configure-le pour utiliser le certificat signé par ta CA et la clé privée.
  - Assure-toi que le serveur web utilise HTTPS (port 443).
  - Redémarre le serveur web.

Étape 6 : Tester la connexion HTTPS depuis les clients

- Sur Client1 et Client2 :
  - Avec un navigateur ou curl, essaie de te connecter à https://IP-serveur (ex : https://192.168.x.x).
  - Vérifie que le certificat est bien reconnu (pas d’erreur de certificat non fiable).
  - Tu peux aussi vérifier que c’est la même CA qui a signé les deux certificats.

Bonus — Vérifications et détails

- Avec la commande openssl s_client -connect IP:443 sur les clients, tu peux voir les détails du certificat.
- Vérifie que la chaîne de certification correspond bien à ta CA.
- Vérifie que les clients font confiance à la CA (pas d’erreur dans les navigateurs ou curl).

En résumé

Étape	**Objectif**
1. Vérifier réseau	Machines communiquent entre elles
2. Créer CA	Autorité de certification créée sur CA-Server
3. Distribuer CA cert	CA reconnue sur tous les serveurs et clients
4. Créer certificats	Certificats signés pour HTTPS-Server1 et 2
5. Configurer HTTPS	Serveurs HTTPS configurés avec certificats
6. Tester connexion	Clients se connectent sans erreur de certificat

Rôle du PC hôte Windows 11 Pro dans l'exercice

1. **Faire tourner les machines virtuelles** Debian via VMware.
   - C'est le logiciel VMware installé sur ton PC qui crée et gère les VM (démarrage, arrêt, configuration réseau).
   - Ton PC doit être allumé et VMware lancé pour que tes VM fonctionnent.
2. **Configurer la connexion réseau des VM**
   - Dans VMware, configurer le réseau des VM pour qu'elles soient sur le même réseau (par exemple réseau interne, bridge, NAT selon ton besoin).
   - S'assurer que les VM puissent communiquer entre elles (ping entre VM).
   - Ce paramétrage se fait dans VMware, pas dans les VM elles-mêmes.
3. **Accéder aux VM si besoin**
   - Tu peux te connecter en SSH ou via la console VMware pour gérer les VM.
   - Depuis Windows, tu peux utiliser un client SSH (ex: PuTTY, Windows Terminal) pour te connecter aux VM.
4. **Transfert de fichiers entre PC hôte et VM (optionnel)**
   - Si tu veux transférer des fichiers (comme les certificats, clés, scripts), tu peux utiliser des partages VMware, SCP via SSH, ou un dossier partagé VMware.

Ce que ton PC hôte ne fait pas dans l'exercice :

- Il ne joue pas le rôle de serveur ou client dans le réseau HTTPS.
- Il ne crée pas les certificats ni ne fait de configuration réseau dans les VM elles-mêmes.
- Il ne participe pas à la communication HTTPS entre les VM.

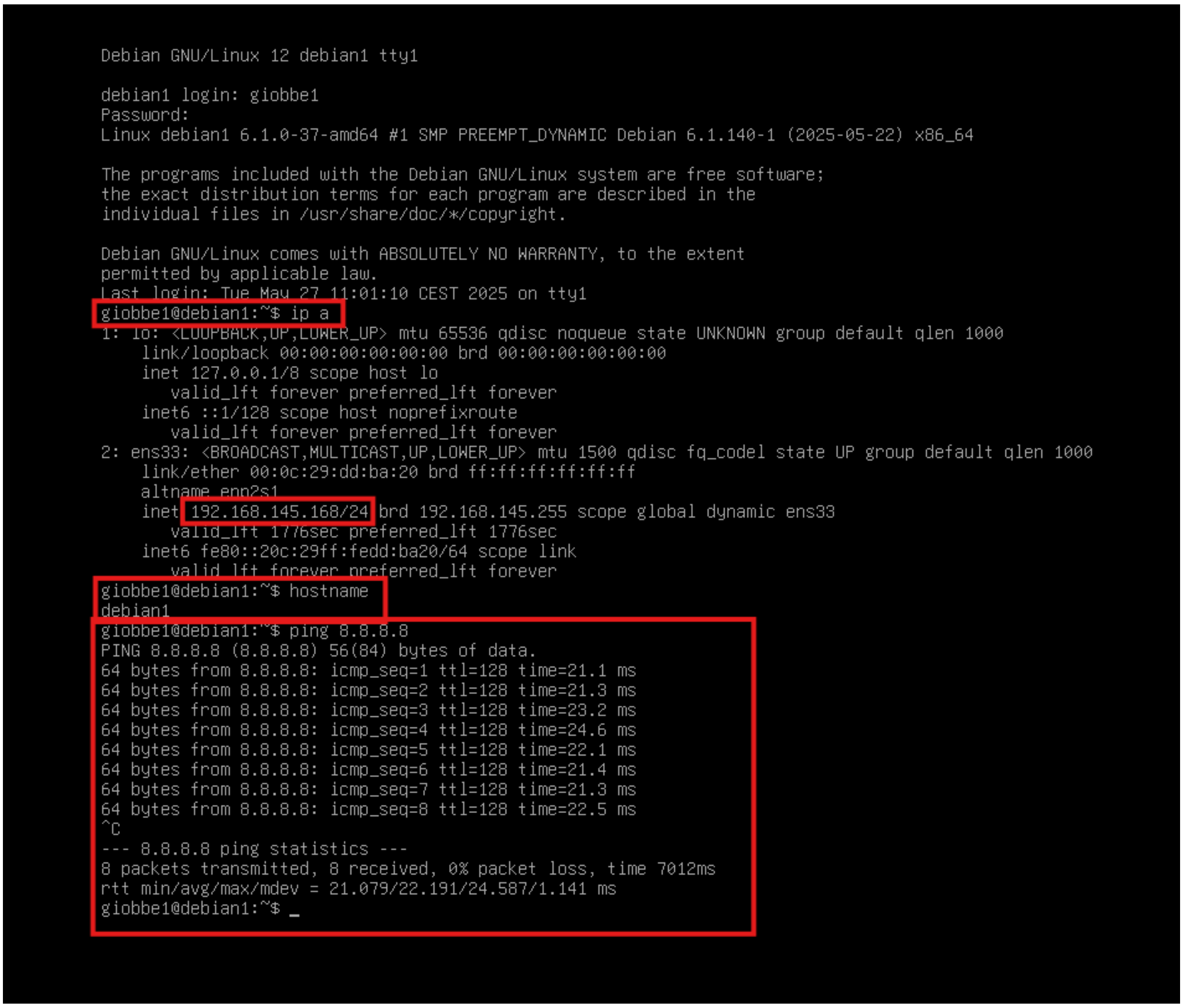
En résumé simple :

PC hôte Windows 11 Pro	**Machines virtuelles Debian**
Gère VMware et la virtualisation	Effectue toutes les actions demandées dans l'exercice (CA, serveurs HTTPS, clients)
Configure le réseau virtuel	Communiquent entre elles via ce réseau
Accède aux VM pour gestion	Hébergent le service HTTPS et font les tests

HTTPS

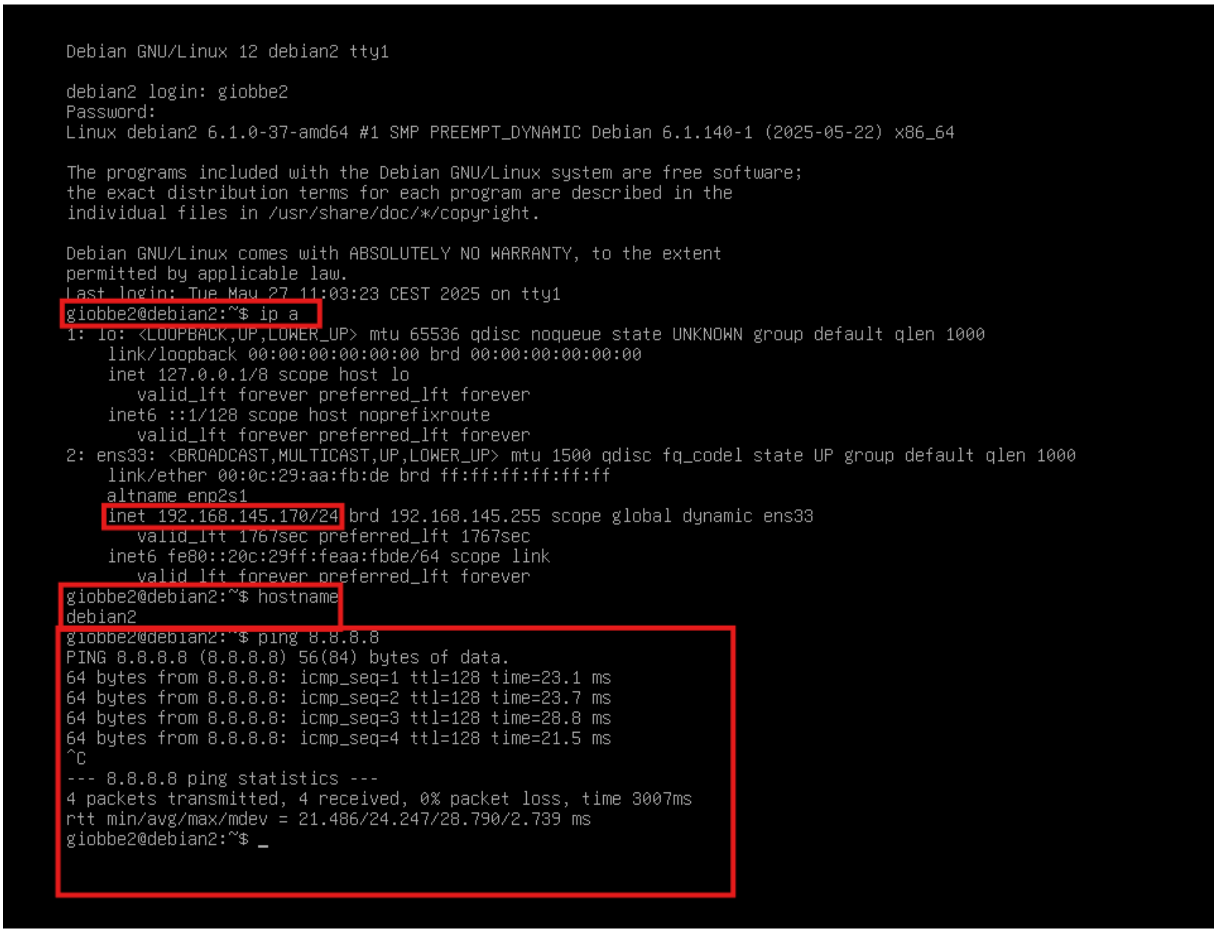
Étape 1 : Vérification réseau et noms/IP

Machine CA-Server (Debian-Console)



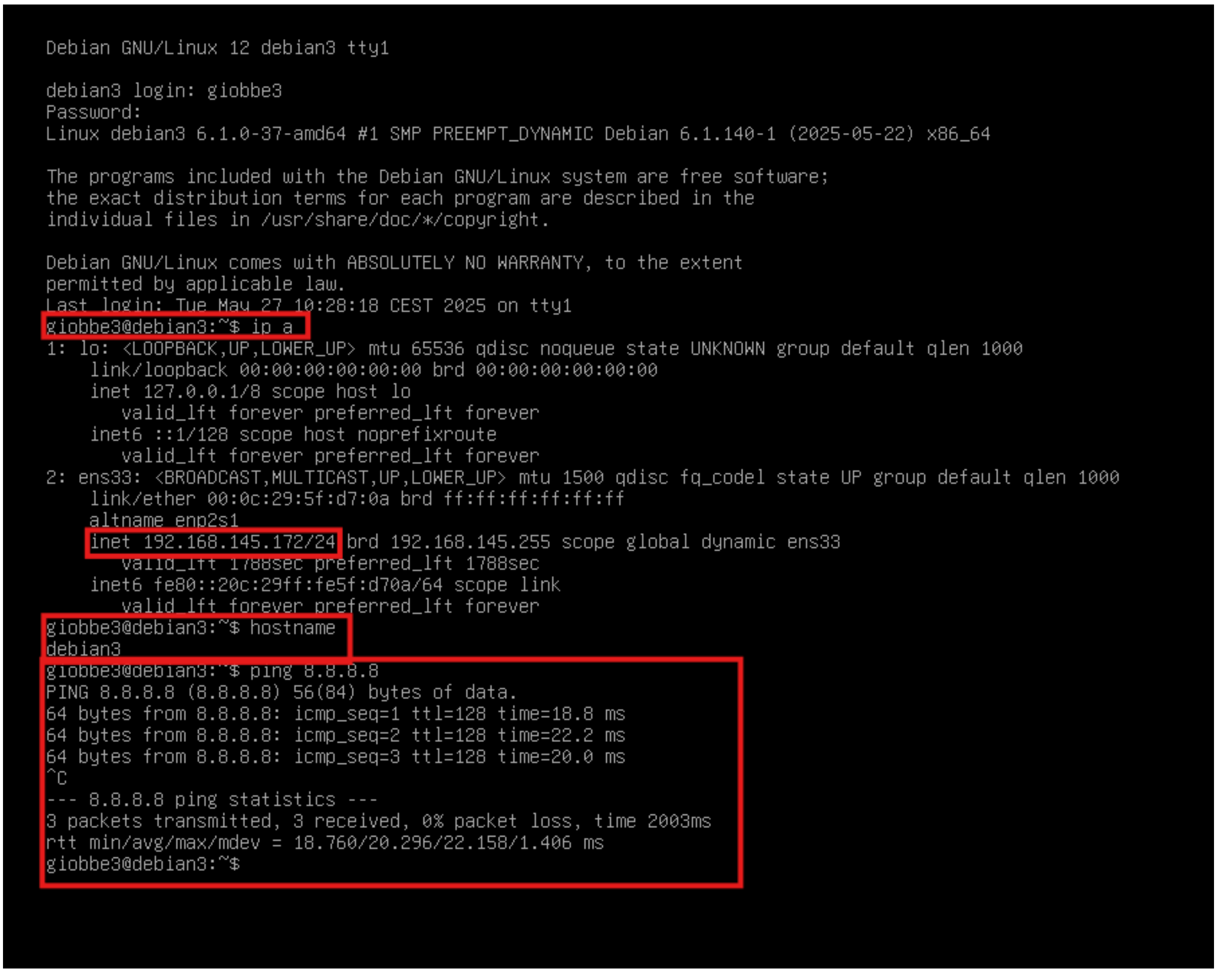
```
Debian GNU/Linux 12 debian1 tty1

debian1 login: giobbe1
Password:
Linux debian1 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 27 11:01:10 CEST 2025 on tty1
giobbe1@debian1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:dd:ba:20 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.145.168/24 brd 192.168.145.255 scope global dynamic ens33
       valid_lft 1776sec preferred_lft 1776sec
    inet6 fe80::20c:29ff:fedd:ba20/64 scope link
       valid_lft forever preferred_lft forever
giobbe1@debian1:~$ hostname
debian1
giobbe1@debian1:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=21.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=21.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=23.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=24.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=22.1 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=128 time=21.4 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=128 time=21.3 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=128 time=22.5 ms
^C
--- 8.8.8.8 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7012ms
rtt min/avg/max/mdev = 21.079/22.191/24.587/1.141 ms
giobbe1@debian1:~$ _
```

Machine HTTPS-Server1 (Debian-Console)



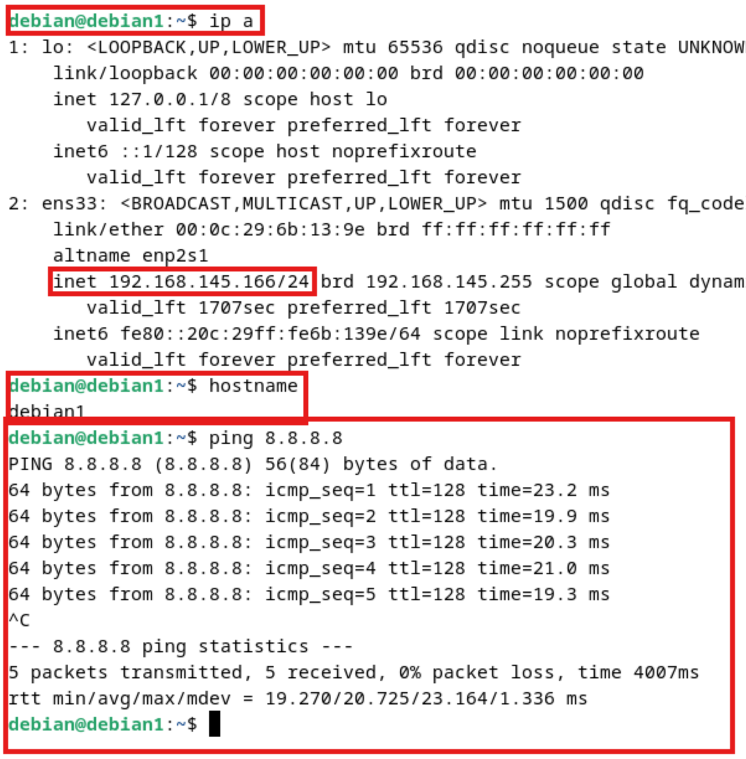
```
Debian GNU/Linux 12 debian2 tty1

debian2 login: giobbe2
Password:
Linux debian2 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 27 11:03:23 CEST 2025 on tty1
giobbe2@debian2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:aa:fb:de brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.145.170/24 brd 192.168.145.255 scope global dynamic ens33
       valid_lft 1767sec preferred_lft 1767sec
    inet6 fe80::20c:29ff:feaa:fbde/64 scope link
       valid_lft forever preferred_lft forever
giobbe2@debian2:~$ hostname
debian2
giobbe2@debian2:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=23.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=23.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=28.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=21.5 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 21.486/24.247/28.790/2.739 ms
giobbe2@debian2:~$ _
```

Machine HTTPS-Server2 (Debian-Console)



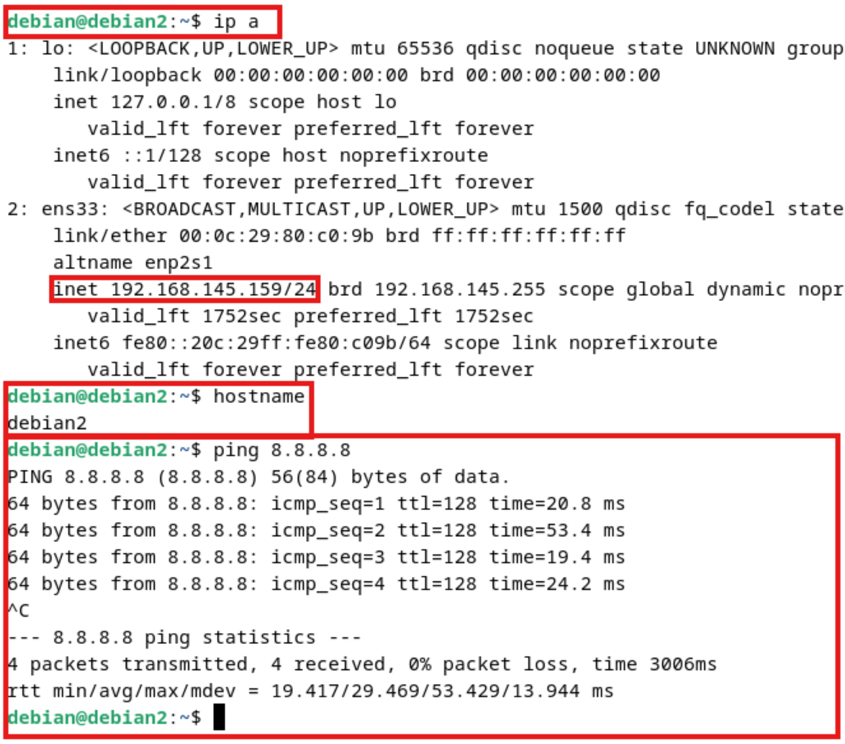
```
Debian GNU/Linux 12 debian3 tty1

debian3 login: giobbe3
Password:
Linux debian3 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 27 10:28:18 CEST 2025 on tty1
giobbe3@debian3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:5f:d7:0a brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.145.172/24 brd 192.168.145.255 scope global dynamic ens33
       valid_lft 1788sec preferred_lft 1788sec
    inet6 fe80::20c:29ff:fe5f:d70a/64 scope link
       valid_lft forever preferred_lft forever
giobbe3@debian3:~$ hostname
debian3
giobbe3@debian3:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=18.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=22.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=20.0 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 18.760/20.296/22.158/1.406 ms
giobbe3@debian3:~$
```

Machine CLIENT-HTTPS1 (Debian-Graphique)



```
debian@debian1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:6b:13:9e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.145.166/24 brd 192.168.145.255 scope global dynamic noprefixroute ens33
       valid_lft 1707sec preferred_lft 1707sec
    inet6 fe80::20c:29ff:fe6b:139e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
debian@debian1:~$ hostname
debian1
debian@debian1:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=23.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=19.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=20.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=128 time=19.3 ms
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 19.270/20.725/23.164/1.336 ms
debian@debian1:~$
```

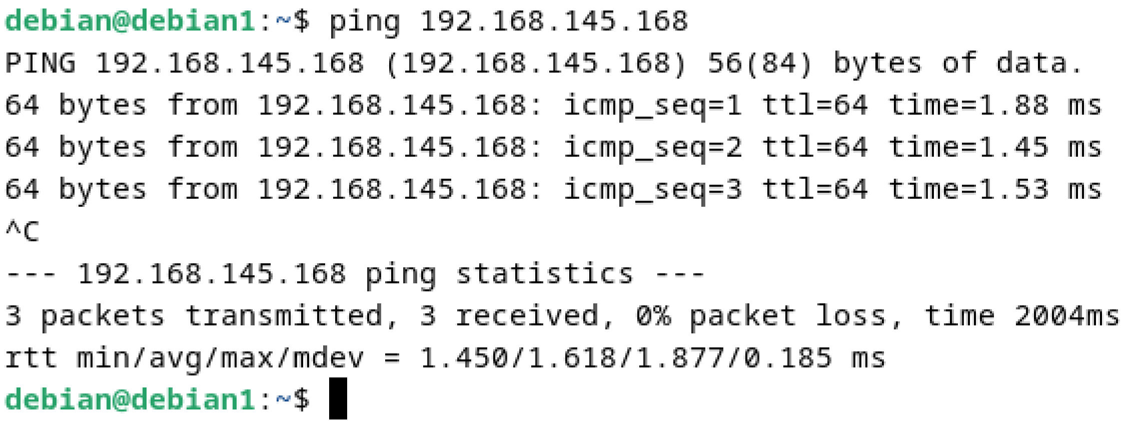
Machine CLIENT-HTTPS2 (Debian-Graphique)



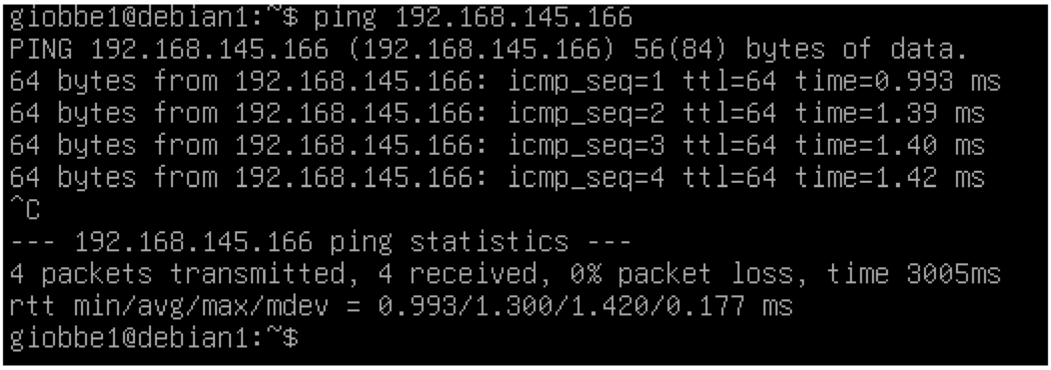
```
debian@debian2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:80:c0:9b brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.145.159/24 brd 192.168.145.255 scope global dynamic noprefixroute ens33
       valid_lft 1752sec preferred_lft 1752sec
    inet6 fe80::20c:29ff:fe80:c09b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
debian@debian2:~$ hostname
debian2
debian@debian2:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=20.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=128 time=53.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=128 time=19.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=128 time=24.2 ms
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 19.417/29.469/53.429/13.944 ms
debian@debian2:~$
```

- Vérifie la communication réseau entre toutes les machines avec la commande ping :
    - Depuis chaque client vers chaque serveur et le CA-Server
    - Depuis chaque serveur vers les autres serveurs et clients

Machine CLIENT-HTTPS1 (Debian-Graphique) → Machine CA-Server (Debian-Console)

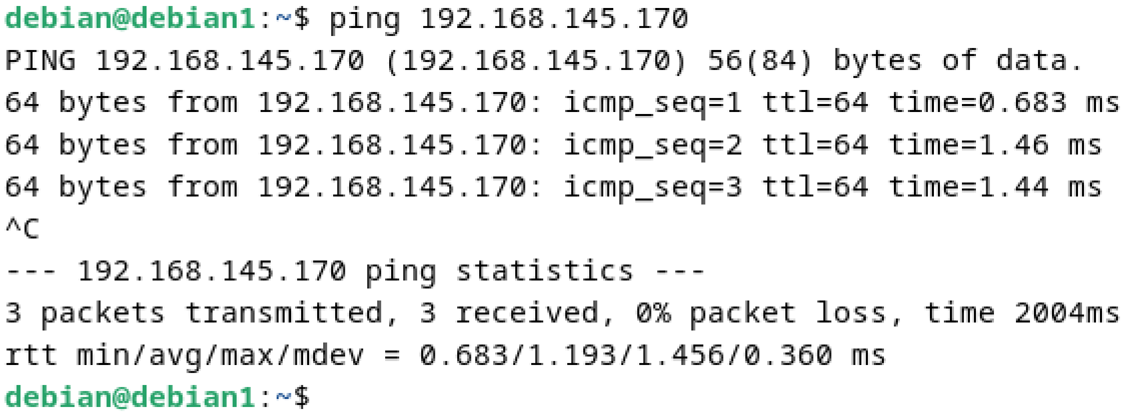


```
debian@debian1:~$ ping 192.168.145.168
PING 192.168.145.168 (192.168.145.168) 56(84) bytes of data.
64 bytes from 192.168.145.168: icmp_seq=1 ttl=64 time=1.88 ms
64 bytes from 192.168.145.168: icmp_seq=2 ttl=64 time=1.45 ms
64 bytes from 192.168.145.168: icmp_seq=3 ttl=64 time=1.53 ms
^C
--- 192.168.145.168 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.450/1.618/1.877/0.185 ms
debian@debian1:~$ 
```

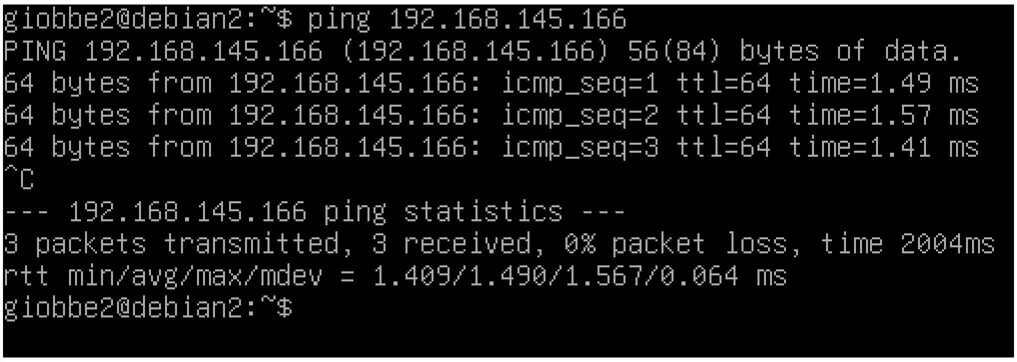


```
giobbe1@debian1:~$ ping 192.168.145.166
PING 192.168.145.166 (192.168.145.166) 56(84) bytes of data.
64 bytes from 192.168.145.166: icmp_seq=1 ttl=64 time=0.993 ms
64 bytes from 192.168.145.166: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 192.168.145.166: icmp_seq=3 ttl=64 time=1.40 ms
64 bytes from 192.168.145.166: icmp_seq=4 ttl=64 time=1.42 ms
^C
--- 192.168.145.166 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.993/1.300/1.420/0.177 ms
giobbe1@debian1:~$
```

Machine CLIENT-HTTPS1 (Debian-Graphique) → Machine HTTPS-Server1 (Debian-Console)

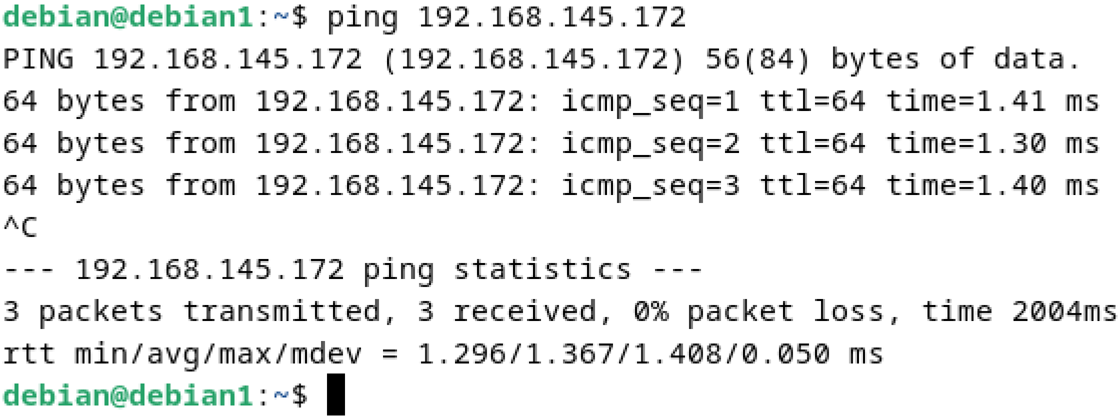


```
debian@debian1:~$ ping 192.168.145.170
PING 192.168.145.170 (192.168.145.170) 56(84) bytes of data.
64 bytes from 192.168.145.170: icmp_seq=1 ttl=64 time=0.683 ms
64 bytes from 192.168.145.170: icmp_seq=2 ttl=64 time=1.46 ms
64 bytes from 192.168.145.170: icmp_seq=3 ttl=64 time=1.44 ms
^C
--- 192.168.145.170 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.683/1.193/1.456/0.360 ms
debian@debian1:~$
```

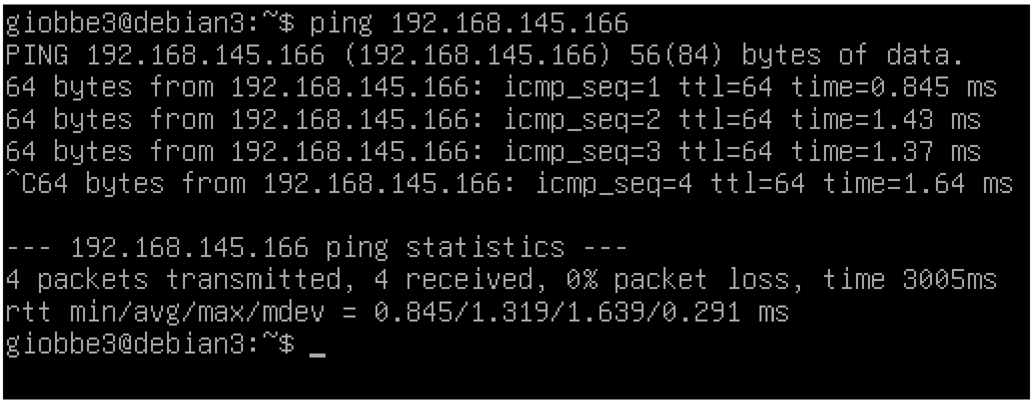


```
giobbe2@debian2:~$ ping 192.168.145.166
PING 192.168.145.166 (192.168.145.166) 56(84) bytes of data.
64 bytes from 192.168.145.166: icmp_seq=1 ttl=64 time=1.49 ms
64 bytes from 192.168.145.166: icmp_seq=2 ttl=64 time=1.57 ms
64 bytes from 192.168.145.166: icmp_seq=3 ttl=64 time=1.41 ms
^C
--- 192.168.145.166 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.409/1.490/1.567/0.064 ms
giobbe2@debian2:~$
```

Machine CLIENT-HTTPS1 (Debian-Graphique) → **Machine HTTPS-Server2 (Debian-Console)**

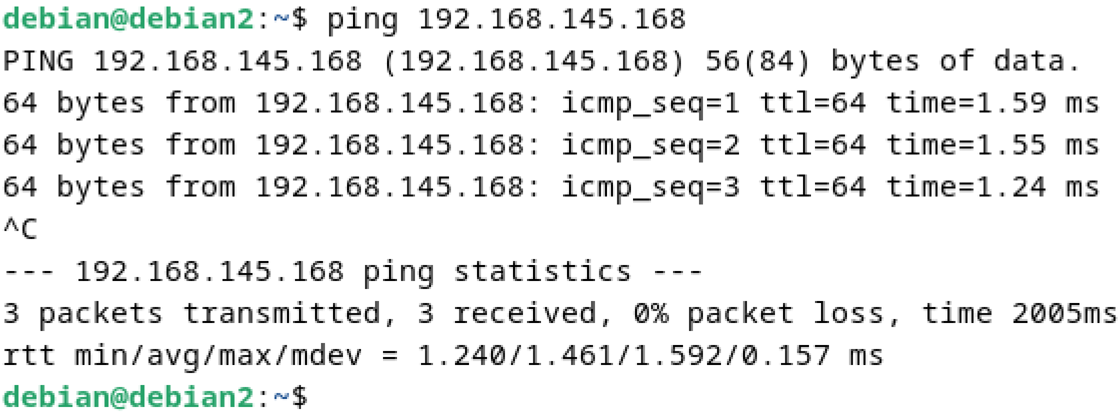


```
debian@debian1:~$ ping 192.168.145.172
PING 192.168.145.172 (192.168.145.172) 56(84) bytes of data.
64 bytes from 192.168.145.172: icmp_seq=1 ttl=64 time=1.41 ms
64 bytes from 192.168.145.172: icmp_seq=2 ttl=64 time=1.30 ms
64 bytes from 192.168.145.172: icmp_seq=3 ttl=64 time=1.40 ms
^C
--- 192.168.145.172 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.296/1.367/1.408/0.050 ms
debian@debian1:~$
```

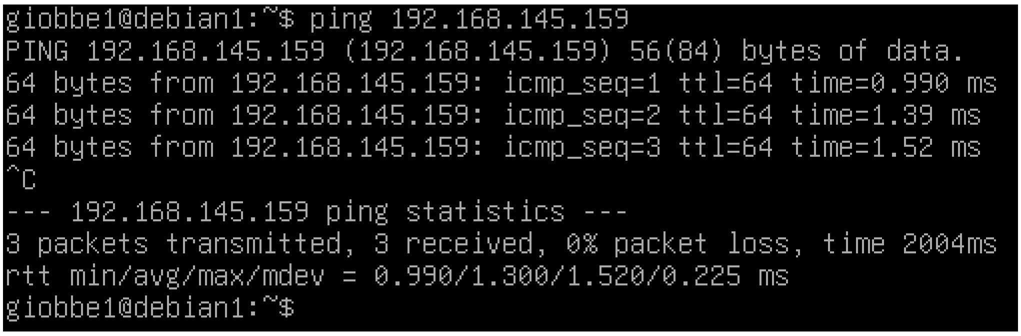


```
giobbe3@debian3:~$ ping 192.168.145.166
PING 192.168.145.166 (192.168.145.166) 56(84) bytes of data.
64 bytes from 192.168.145.166: icmp_seq=1 ttl=64 time=0.845 ms
64 bytes from 192.168.145.166: icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from 192.168.145.166: icmp_seq=3 ttl=64 time=1.37 ms
^C64 bytes from 192.168.145.166: icmp_seq=4 ttl=64 time=1.64 ms

--- 192.168.145.166 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.845/1.319/1.639/0.291 ms
giobbe3@debian3:~$ _
```

Machine CLIENT-HTTPS2 (Debian-Graphique) → Machine CA-Server (Debian-Console)

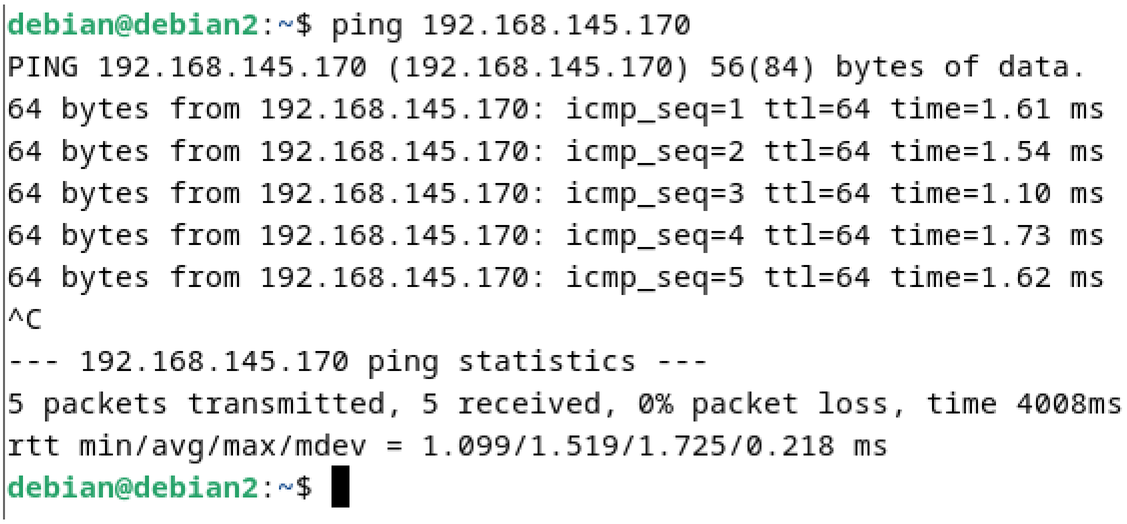


```
debian@debian2:~$ ping 192.168.145.168
PING 192.168.145.168 (192.168.145.168) 56(84) bytes of data.
64 bytes from 192.168.145.168: icmp_seq=1 ttl=64 time=1.59 ms
64 bytes from 192.168.145.168: icmp_seq=2 ttl=64 time=1.55 ms
64 bytes from 192.168.145.168: icmp_seq=3 ttl=64 time=1.24 ms
^C
--- 192.168.145.168 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 1.240/1.461/1.592/0.157 ms
debian@debian2:~$
```

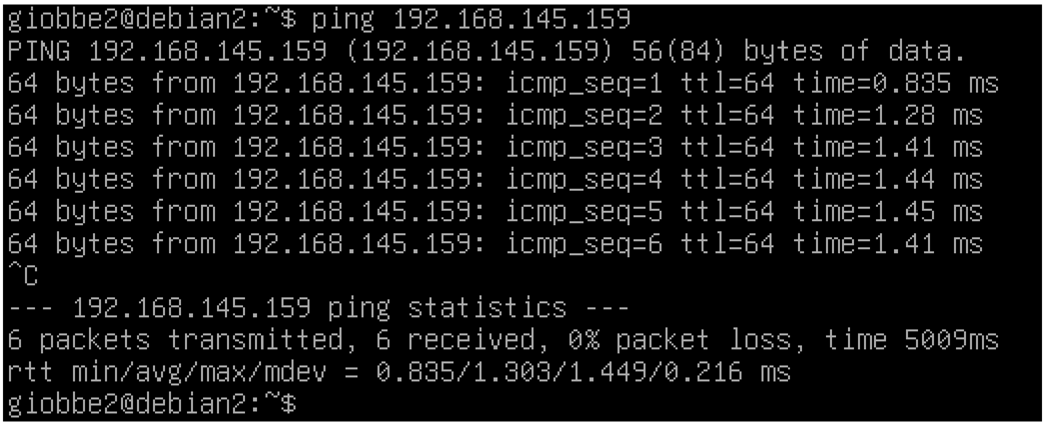


```
giobbe1@debian1:~$ ping 192.168.145.159
PING 192.168.145.159 (192.168.145.159) 56(84) bytes of data.
64 bytes from 192.168.145.159: icmp_seq=1 ttl=64 time=0.990 ms
64 bytes from 192.168.145.159: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 192.168.145.159: icmp_seq=3 ttl=64 time=1.52 ms
^C
--- 192.168.145.159 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.990/1.300/1.520/0.225 ms
giobbe1@debian1:~$
```

Machine CLIENT-HTTPS2 (Debian-Graphique) → Machine HTTPS-Server1 (Debian-Console)

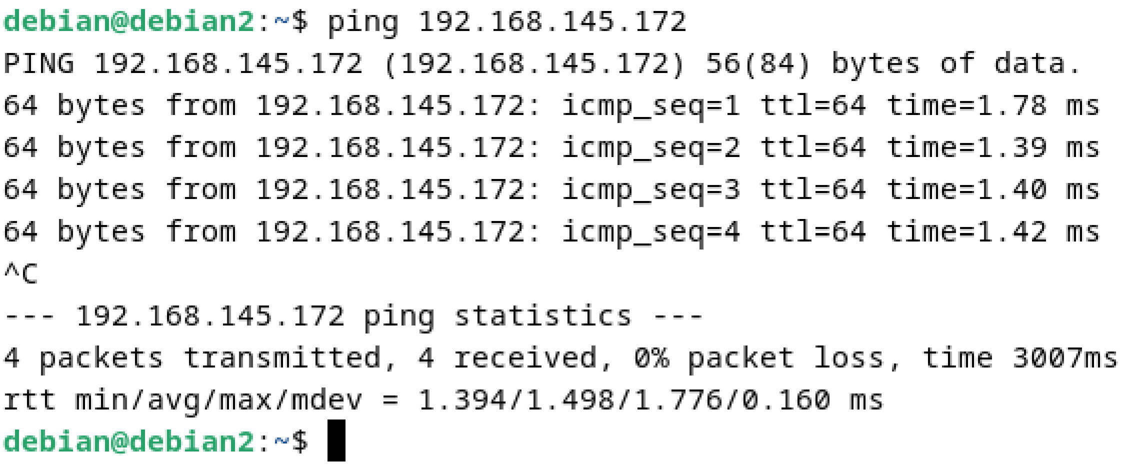


```
debian@debian2:~$ ping 192.168.145.170
PING 192.168.145.170 (192.168.145.170) 56(84) bytes of data.
64 bytes from 192.168.145.170: icmp_seq=1 ttl=64 time=1.61 ms
64 bytes from 192.168.145.170: icmp_seq=2 ttl=64 time=1.54 ms
64 bytes from 192.168.145.170: icmp_seq=3 ttl=64 time=1.10 ms
64 bytes from 192.168.145.170: icmp_seq=4 ttl=64 time=1.73 ms
64 bytes from 192.168.145.170: icmp_seq=5 ttl=64 time=1.62 ms
^C
--- 192.168.145.170 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 1.099/1.519/1.725/0.218 ms
debian@debian2:~$
```

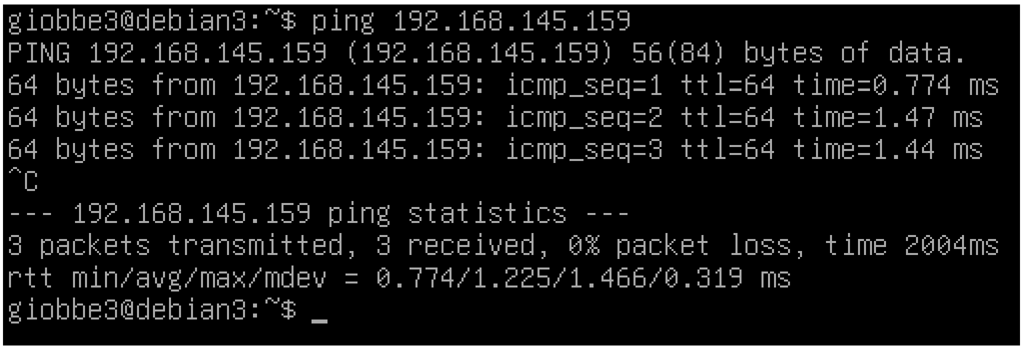


```
giobbe2@debian2:~$ ping 192.168.145.159
PING 192.168.145.159 (192.168.145.159) 56(84) bytes of data.
64 bytes from 192.168.145.159: icmp_seq=1 ttl=64 time=0.835 ms
64 bytes from 192.168.145.159: icmp_seq=2 ttl=64 time=1.28 ms
64 bytes from 192.168.145.159: icmp_seq=3 ttl=64 time=1.41 ms
64 bytes from 192.168.145.159: icmp_seq=4 ttl=64 time=1.44 ms
64 bytes from 192.168.145.159: icmp_seq=5 ttl=64 time=1.45 ms
64 bytes from 192.168.145.159: icmp_seq=6 ttl=64 time=1.41 ms
^C
--- 192.168.145.159 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 0.835/1.303/1.449/0.216 ms
giobbe2@debian2:~$
```

Machine CLIENT-HTTPS2 (Debian-Graphique) → Machine HTTPS-Server1 (Debian-Console)



```
debian@debian2:~$ ping 192.168.145.172
PING 192.168.145.172 (192.168.145.172) 56(84) bytes of data.
64 bytes from 192.168.145.172: icmp_seq=1 ttl=64 time=1.78 ms
64 bytes from 192.168.145.172: icmp_seq=2 ttl=64 time=1.39 ms
64 bytes from 192.168.145.172: icmp_seq=3 ttl=64 time=1.40 ms
64 bytes from 192.168.145.172: icmp_seq=4 ttl=64 time=1.42 ms
^C
--- 192.168.145.172 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.394/1.498/1.776/0.160 ms
debian@debian2:~$
```



```
giobbe3@debian3:~$ ping 192.168.145.159
PING 192.168.145.159 (192.168.145.159) 56(84) bytes of data.
64 bytes from 192.168.145.159: icmp_seq=1 ttl=64 time=0.774 ms
64 bytes from 192.168.145.159: icmp_seq=2 ttl=64 time=1.47 ms
64 bytes from 192.168.145.159: icmp_seq=3 ttl=64 time=1.44 ms
^C
--- 192.168.145.159 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.774/1.225/1.466/0.319 ms
giobbe3@debian3:~$ _
```

Étape 2 : Création de l'Autorité de Certification (CA) sur CA-Server

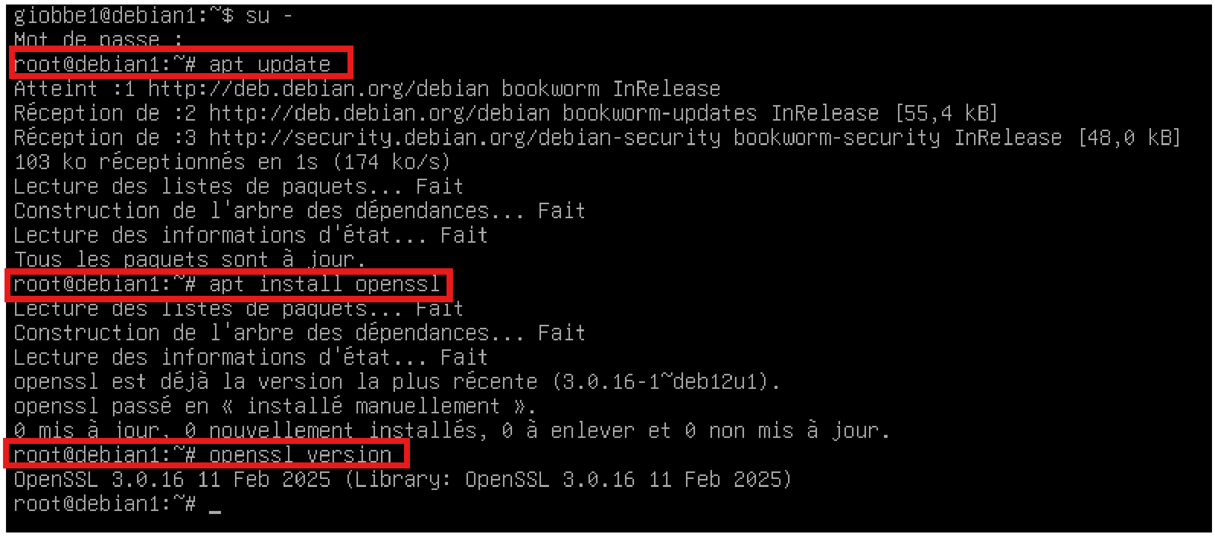
S'assurer qu'OpenSSL est installé

Sur la machine CA-Server :

- sudo apt update

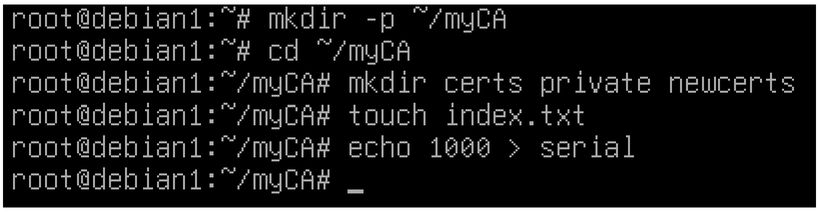
- sudo apt install openssl

Apres vérifie la version installée : openssl version



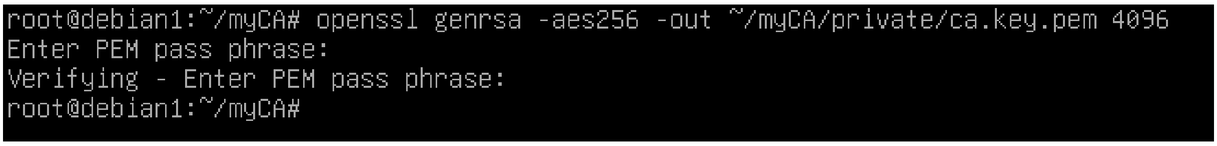
```
giobbe1@debian1:~$ su -
Mot de passe :
root@debian1:~# apt update
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Réception de :2 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Réception de :3 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
103 ko réceptionnés en 1s (174 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@debian1:~# apt install openssl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
openssl est déjà la version la plus récente (3.0.16-1~deb12u1).
openssl passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
root@debian1:~# openssl version
OpenSSL 3.0.16 11 Feb 2025 (Library: OpenSSL 3.0.16 11 Feb 2025)
root@debian1:~# _
```

Créer l'arborescence pour ta CA



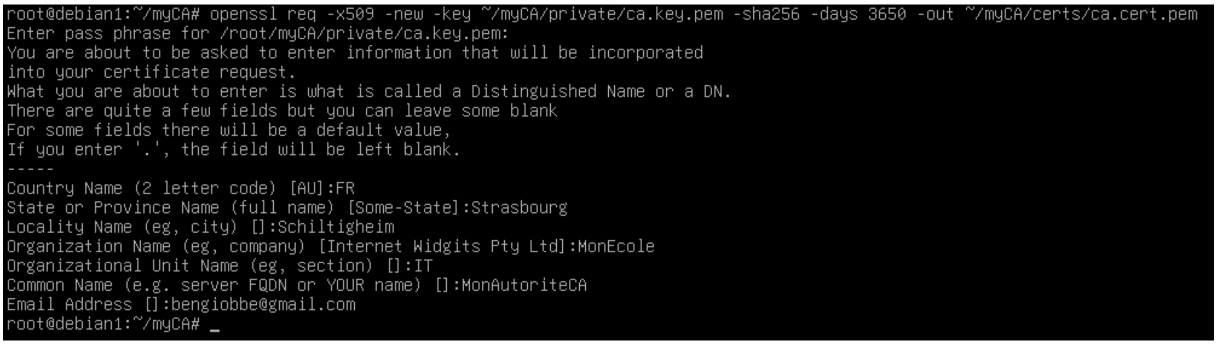
```
root@debian1:~# mkdir -p ~/myCA
root@debian1:~# cd ~/myCA
root@debian1:~/myCA# mkdir certs private newcerts
root@debian1:~/myCA# touch index.txt
root@debian1:~/myCA# echo 1000 > serial
root@debian1:~/myCA# _
```

Générer la clé privée de la CA



```
root@debian1:~/myCA# openssl genrsa -aes256 -out ~/myCA/private/ca.key.pem 4096
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
root@debian1:~/myCA#
```

Créer le certificat racine auto-signé



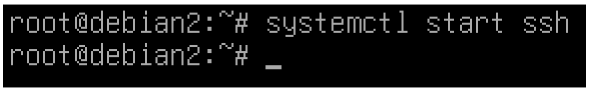
```
root@debian1:~/myCA# openssl req -x509 -new -key ~/myCA/private/ca.key.pem -sha256 -days 3650 -out ~/myCA/certs/ca.cert.pem
Enter pass phrase for /root/myCA/private/ca.key.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Strasbourg
Locality Name (eg, city) []:Schiltigheim
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MonEcole
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:MonAutoriteCA
Email Address []:bengiobbe@gmail.com
root@debian1:~/myCA# _
```

Installation SSH sur la Machine HTTPS-SERVER1

sudo apt update

sudo apt install openssh-server

Démarre le service SSH : sudo systemctl start ssh



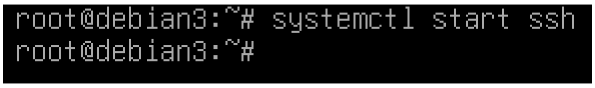
```
root@debian2:~# systemctl start ssh
root@debian2:~# _
```

Installation SSH sur la Machine HTTPS-SERVER2

sudo apt update

sudo apt install openssh-server

Démarre le service SSH : sudo systemctl start ssh



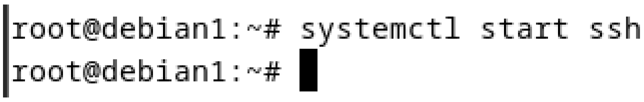
```
root@debian3:~# systemctl start ssh
root@debian3:~#
```

Installation SSH sur la Machine CLIENT HTTPS1

sudo apt update

sudo apt install openssh-server

Démarre le service SSH : sudo systemctl start ssh



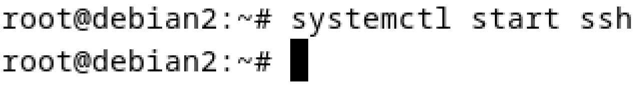
```
root@debian1:~# systemctl start ssh
root@debian1:~# █
```

Installation SSH sur la Machine CLIENT HTTPS2

sudo apt update

sudo apt install openssh-server

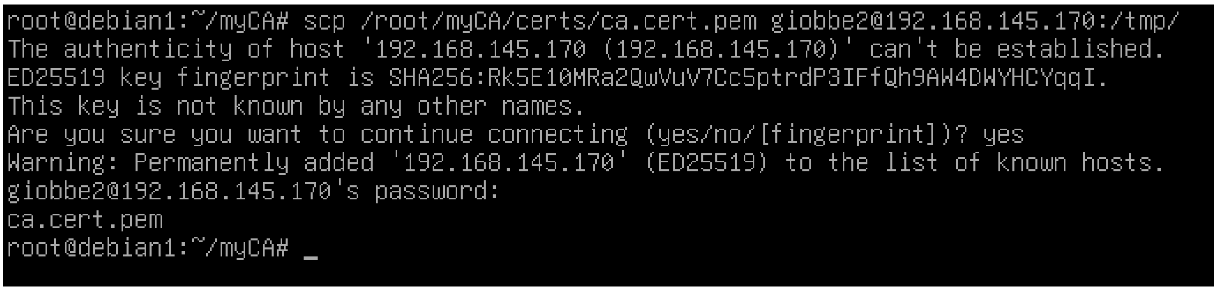
Démarre le service SSH : sudo systemctl start ssh



```
root@debian2:~# systemctl start ssh
root@debian2:~# █
```

Copier le certificat racine CA sur chaque machine distante

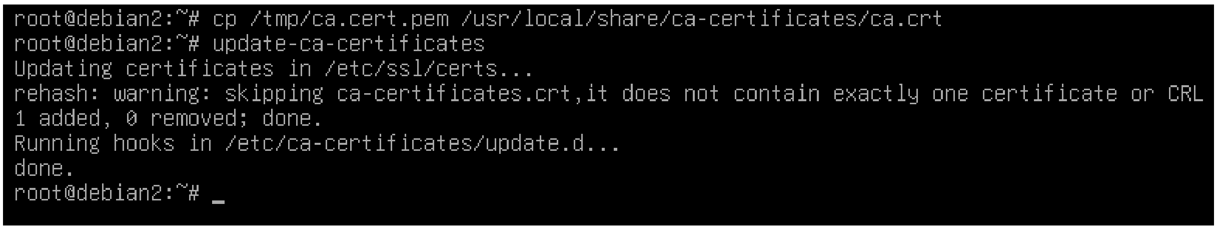
CA VERS HTTPS1



```
root@debian1:~/myCA# scp /root/myCA/certs/ca.cert.pem giobbe2@192.168.145.170:/tmp/
The authenticity of host '192.168.145.170 (192.168.145.170)' can't be established.
ED25519 key fingerprint is SHA256:Rk5E10MRa2QwVuV7Cc5ptrdP3IFfQh9AW4DWYHCYqqI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.145.170' (ED25519) to the list of known hosts.
giobbe2@192.168.145.170's password:
ca.cert.pem
root@debian1:~/myCA# _
```

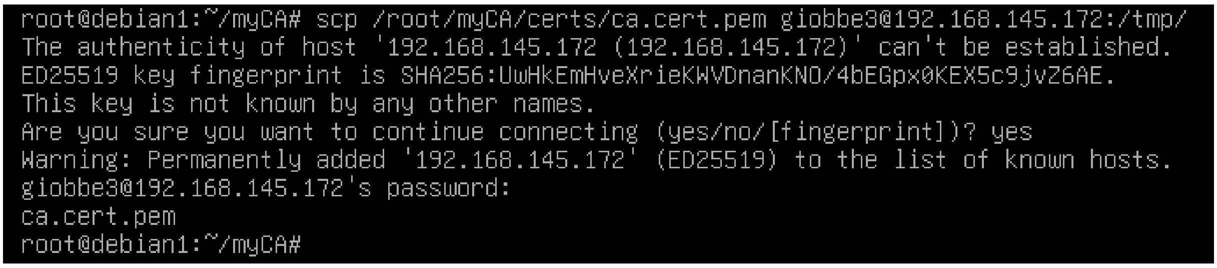
HTTPS1

Maintenant, connecte-toi sur cette machine distante (via SSH ou console) et installe le certificat racine avec ces commandes :



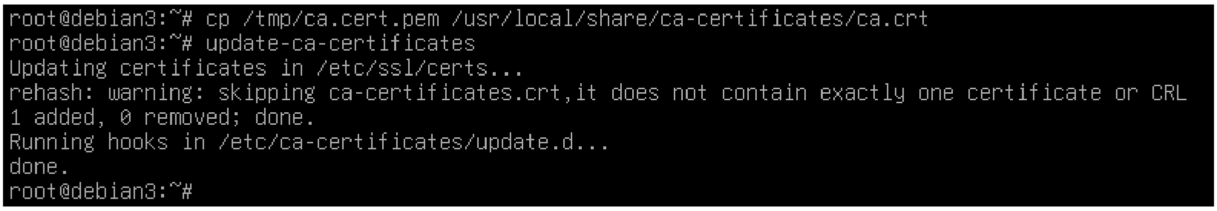
```
root@debian2:~# cp /tmp/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
root@debian2:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian2:~# _
```

CA VERS HTTPS2



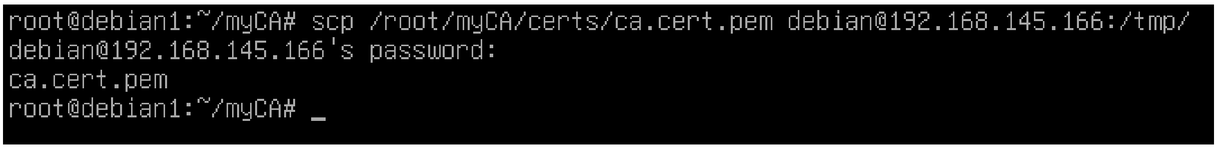
```
root@debian1:~/myCA# scp /root/myCA/certs/ca.cert.pem giobbe3@192.168.145.172:/tmp/
The authenticity of host '192.168.145.172 (192.168.145.172)' can't be established.
ED25519 key fingerprint is SHA256:UwHkEmHveXrieKWVDnanKNO/4bEGpx0KEX5c9jvZ6AE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.145.172' (ED25519) to the list of known hosts.
giobbe3@192.168.145.172's password:
ca.cert.pem
root@debian1:~/myCA#
```

HTTPS2



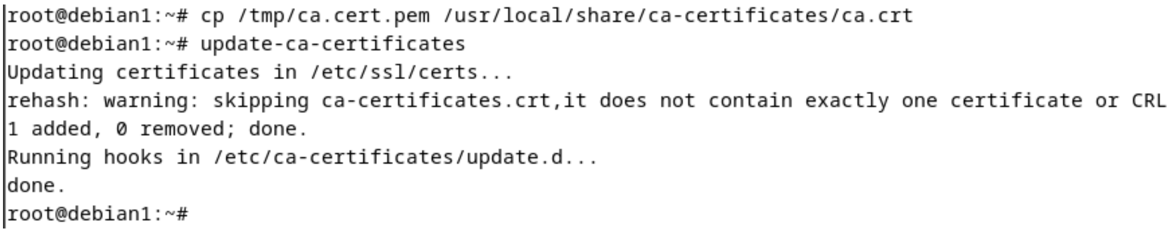
```
root@debian3:~# cp /tmp/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
root@debian3:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian3:~#
```

CA VERS CLIENT HTTPS1



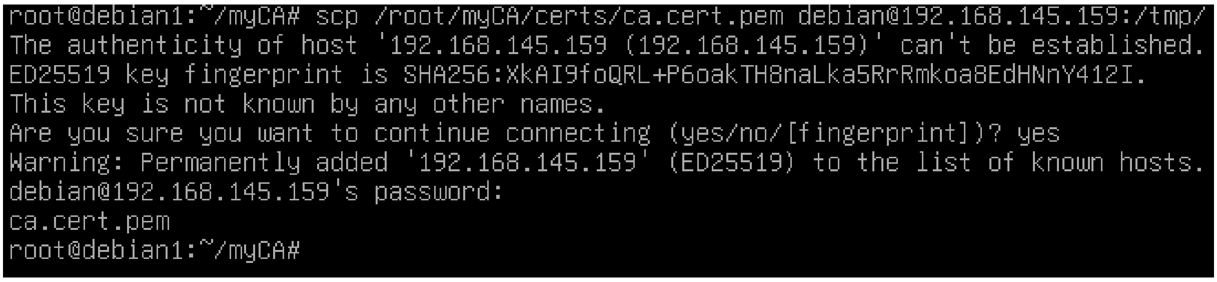
```
root@debian1:~/myCA# scp /root/myCA/certs/ca.cert.pem debian@192.168.145.166:/tmp/
debian@192.168.145.166's password:
ca.cert.pem
root@debian1:~/myCA# _
```

CLIENT HTTPS1



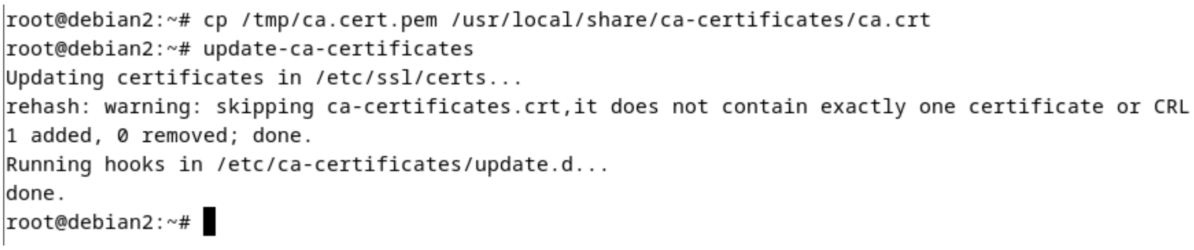
```
root@debian1:~# cp /tmp/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
root@debian1:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian1:~#
```

CA VERS CLIENT HTTPS2



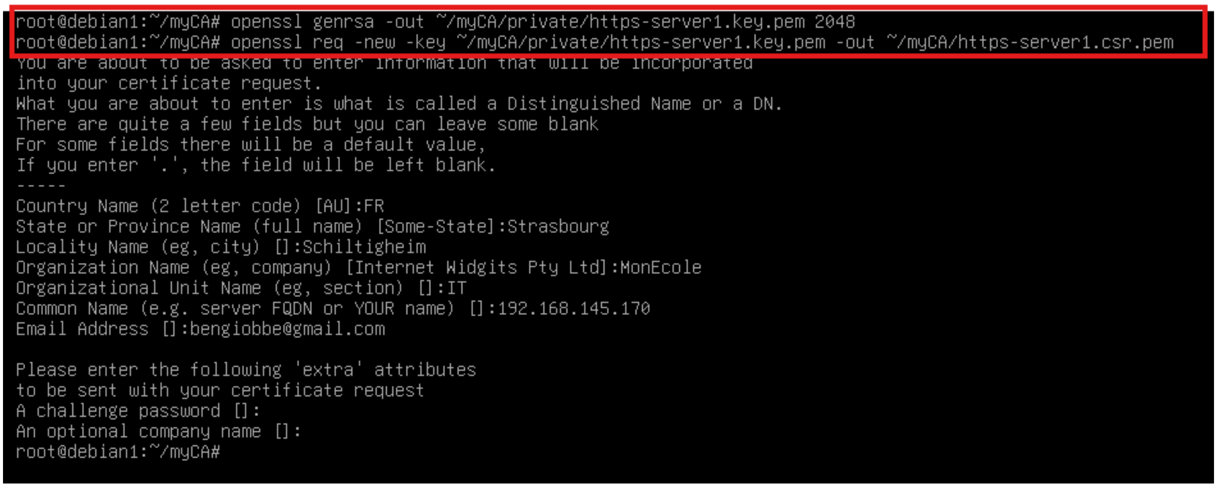
```
root@debian1:~/myCA# scp /root/myCA/certs/ca.cert.pem debian@192.168.145.159:/tmp/
The authenticity of host '192.168.145.159 (192.168.145.159)' can't be established.
ED25519 key fingerprint is SHA256:XkAI9foQRL+P6oakTH8naLka5RrRmkoa8EdHNnY412I.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.145.159' (ED25519) to the list of known hosts.
debian@192.168.145.159's password:
ca.cert.pem
root@debian1:~/myCA#
```

CLIENT HTTPS2



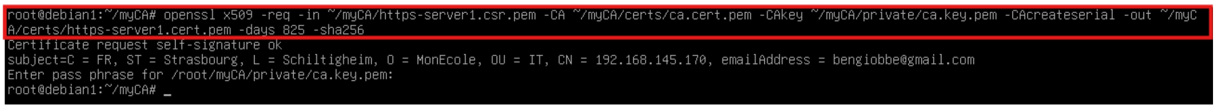
```
root@debian2:~# cp /tmp/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
root@debian2:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian2:~# █
```

Créer une clé privée et une demande de signature (CSR) pour HTTPS-Server1



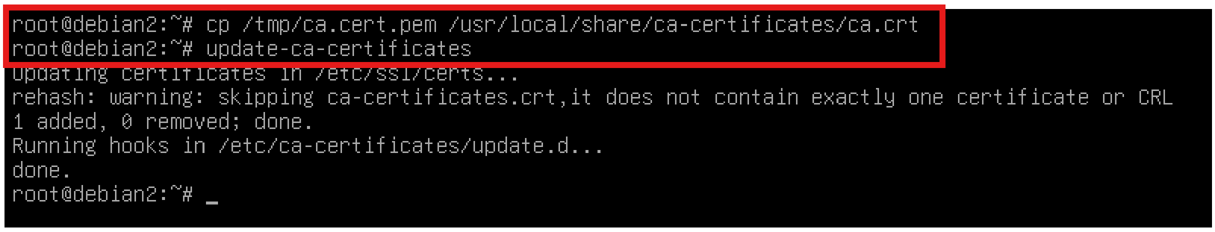
```
root@debian1:~/myCA# openssl genrsa -out ~/myCA/private/https-server1.key.pem 2048
root@debian1:~/myCA# openssl req -new -key ~/myCA/private/https-server1.key.pem -out ~/myCA/https-server1.csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Strasbourg
Locality Name (eg, city) []:Schiltigheim
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MonEcole
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:192.168.145.170
Email Address []:bengiobbe@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@debian1:~/myCA#
```

Signer la demande (CSR) avec ta CA pour créer le certificat



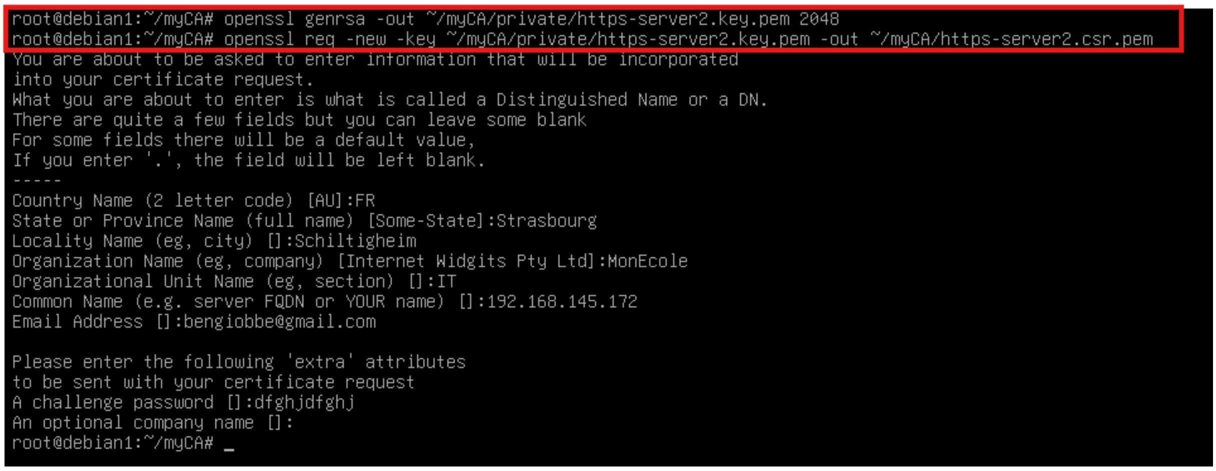
```
root@debian1:~/myCA# openssl x509 -req -in ~/myCA/https-server1.csr.pem -CA ~/myCA/certs/ca.cert.pem -CAkey ~/myCA/private/ca.key.pem -CAcreateserial -out ~/myC
A/certs/https-server1.cert.pem -days 825 -sha256
Certificate request self-signature ok
subject=C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = 192.168.145.170, emailAddress = bengiobbe@gmail.com
Enter pass phrase for /root/myCA/private/ca.key.pem:
root@debian1:~/myCA# _
```

Installer le certificat racine CA sur la machine HTTPS-Server1



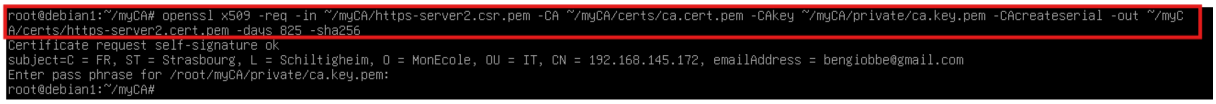
```
root@debian2:~# cp /tmp/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
root@debian2:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian2:~# _
```

Créer une clé privée et une demande de signature (CSR) pour HTTPS-Server2



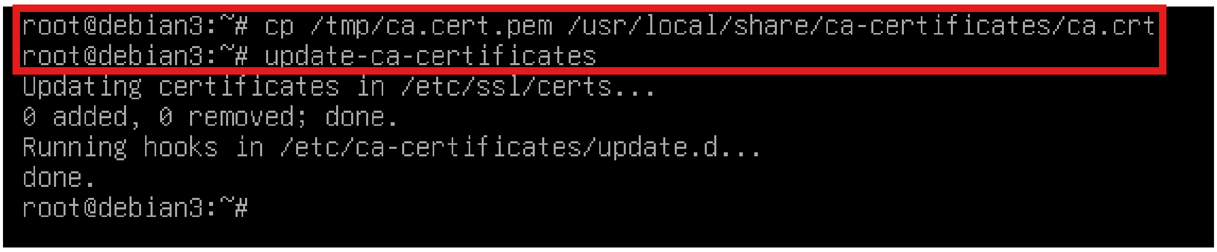
```
root@debian1:~/myCA# openssl genrsa -out ~/myCA/private/https-server2.key.pem 2048
root@debian1:~/myCA# openssl req -new -key ~/myCA/private/https-server2.key.pem -out ~/myCA/https-server2.csr.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Strasbourg
Locality Name (eg, city) []:Schiltigheim
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MonEcole
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:192.168.145.172
Email Address []:bengiobbe@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:dfghjdfghj
An optional company name []:
root@debian1:~/myCA# _
```

Signer la demande (CSR) avec ta CA pour créer le certificat



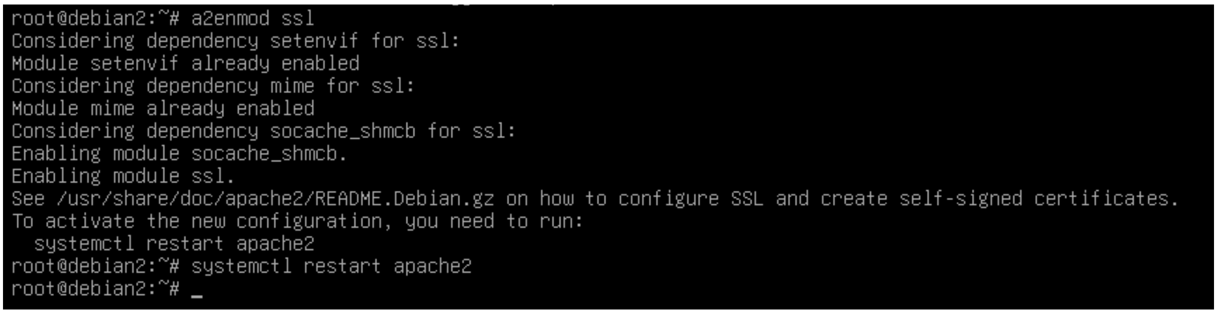
```
root@debian1:~/myCA# openssl x509 -req -in ~/myCA/https-server2.csr.pem -CA ~/myCA/certs/ca.cert.pem -CAkey ~/myCA/private/ca.key.pem -CAcreateserial -out ~/myC
A/certs/https-server2.cert.pem -days 825 -sha256
Certificate request self-signature ok
subject=C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = 192.168.145.172, emailAddress = bengiobbe@gmail.com
Enter pass phrase for /root/myCA/private/ca.key.pem:
root@debian1:~/myCA#
```

Installer le certificat racine CA sur la machine HTTPS-Server2



```
root@debian3:~# cp /tmp/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
root@debian3:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian3:~#
```

Installation Apache2 et le module SSL sur le HTTPS-Server1



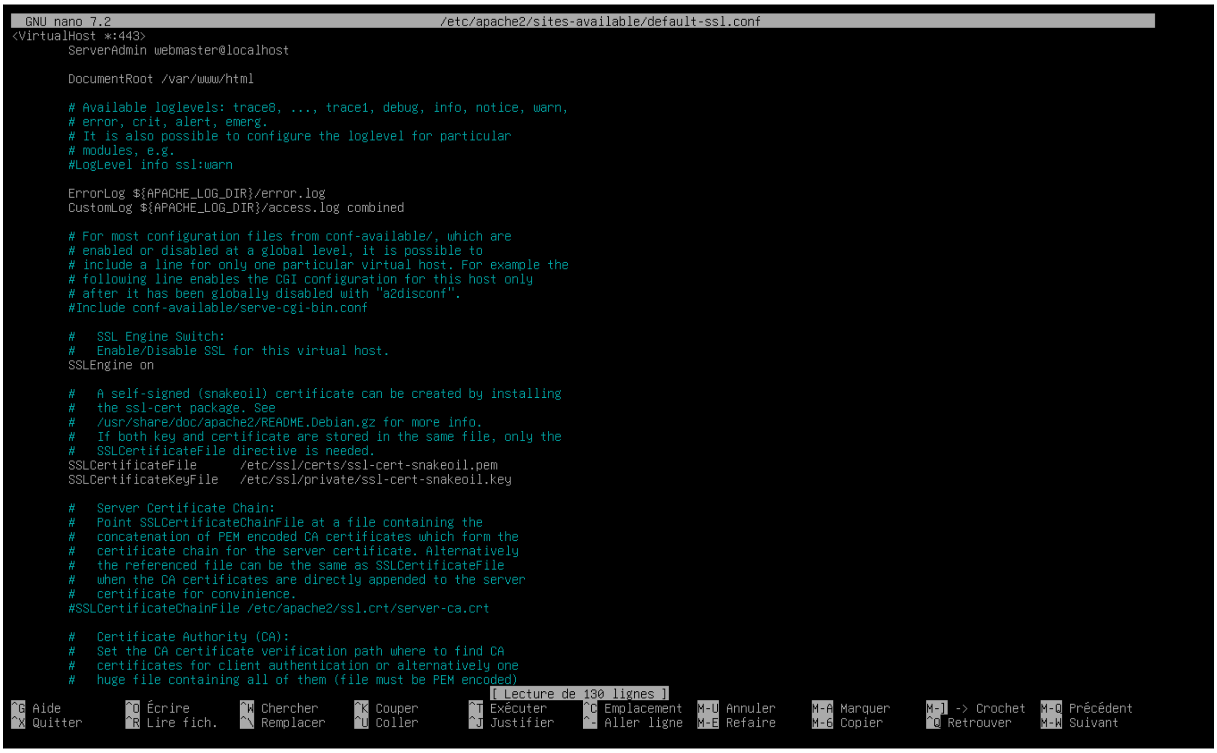
```
root@debian2:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian2:~# systemctl restart apache2
root@debian2:~# _
```

Configurer Apache pour activer HTTPS

Modifier le fichier de configuration HTTPS

Tape cette commande pour ouvrir ce fichier avec un éditeur de texte (nano) :

nano /etc/apache2/sites-available/default-ssl.conf



```
  GNU nano 7.2                                    /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:443>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
        SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key

        #   Server Certificate Chain:
        #   Point SSLCertificateChainFile at a file containing the
        #   concatenation of PEM encoded CA certificates which form the
        #   certificate chain for the server certificate. Alternatively
        #   the referenced file can be the same as SSLCertificateFile
        #   when the CA certificates are directly appended to the server
        #   certificate for convinience.
        #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

        #   Certificate Authority (CA):
        #   Set the CA certificate verification path where to find CA
        #   certificates for client authentication or alternatively one
        #   huge file containing all of them (file must be PEM encoded)
                                                              [ Lecture de 130 lignes ]
^G Aide        ^O Écrire      ^W Chercher    ^K Couper      ^T Exécuter    ^C Emplacement M-U Annuler     M-A Marquer     M-] -> Crochet  M-Q Précédent
^X Quitter     ^R Lire fich.  ^\ Remplacer   ^U Coller      ^J Justifier   ^- Aller ligne M-E Refaire     M-6 Copier      ^Q Retrouver    M-W Suivant
```

Dans le fichier /etc/apache2/sites-available/default-ssl.conf, il faut remplacer :

SSLCertificateFile   /etc/ssl/certs/ssl-cert-snakeoil.pem

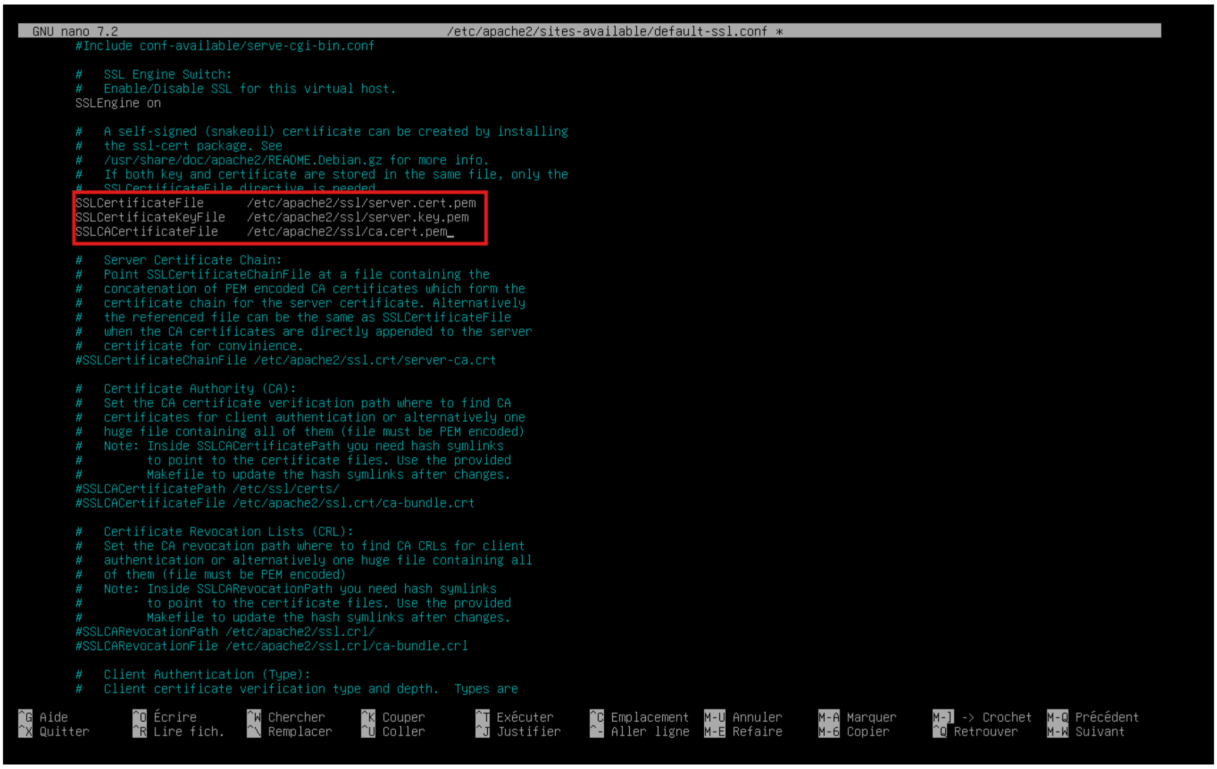
SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key

Par

SSLCertificateFile   /etc/apache2/ssl/server.cert.pem

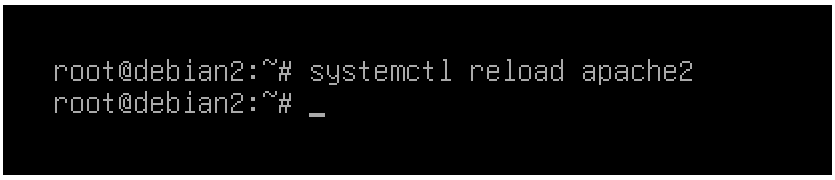
SSLCertificateKeyFile   /etc/apache2/ssl/server.key.pem

SSLCACertificateFile   /etc/apache2/ssl/ca.cert.pem



```
  GNU nano 7.2                                                  /etc/apache2/sites-available/default-ssl.conf *
          #Include conf-available/serve-cgi-bin.conf

          #   SSL Engine Switch:
          #   Enable/Disable SSL for this virtual host.
          SSLEngine on

          #   A self-signed (snakeoil) certificate can be created by installing
          #   the ssl-cert package. See
          #   /usr/share/doc/apache2/README.Debian.gz for more info.
          #   If both key and certificate are stored in the same file, only the
          #   SSLCertificateFile directive is needed.
          SSLCertificateFile      /etc/apache2/ssl/server.cert.pem
          SSLCertificateKeyFile   /etc/apache2/ssl/server.key.pem
          SSLCACertificateFile    /etc/apache2/ssl/ca.cert.pem_

          #   Server Certificate Chain:
          #   Point SSLCertificateChainFile at a file containing the
          #   concatenation of PEM encoded CA certificates which form the
          #   certificate chain for the server certificate. Alternatively
          #   the referenced file can be the same as SSLCertificateFile
          #   when the CA certificates are directly appended to the server
          #   certificate for convinience.
          #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

          #   Certificate Authority (CA):
          #   Set the CA certificate verification path where to find CA
          #   certificates for client authentication or alternatively one
          #   huge file containing all of them (file must be PEM encoded)
          #   Note: Inside SSLCACertificatePath you need hash symlinks
          #         to point to the certificate files. Use the provided
          #         Makefile to update the hash symlinks after changes.
          #SSLCACertificatePath /etc/ssl/certs/
          #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

          #   Certificate Revocation Lists (CRL):
          #   Set the CA revocation path where to find CA CRLs for client
          #   authentication or alternatively one huge file containing all
          #   of them (file must be PEM encoded)
          #   Note: Inside SSLCARevocationPath you need hash symlinks
          #         to point to the certificate files. Use the provided
          #         Makefile to update the hash symlinks after changes.
          #SSLCARevocationPath /etc/apache2/ssl.crl/
          #SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

          #   Client Authentication (Type):
          #   Client certificate verification type and depth.  Types are
^G Aide       ^O Écrire     ^W Chercher   ^K Couper     ^T Exécuter   ^C Emplacement M-U Annuler   M-A Marquer   M-] -> Crochet M-Q Précédent
^X Quitter    ^R Lire fich. ^\ Remplacer  ^U Coller     ^J Justifier  ^- Aller ligne M-E Refaire   M-6 Copier    ^Q Retrouver   M-W Suivant
```

Ensuite, tu relances Apache :

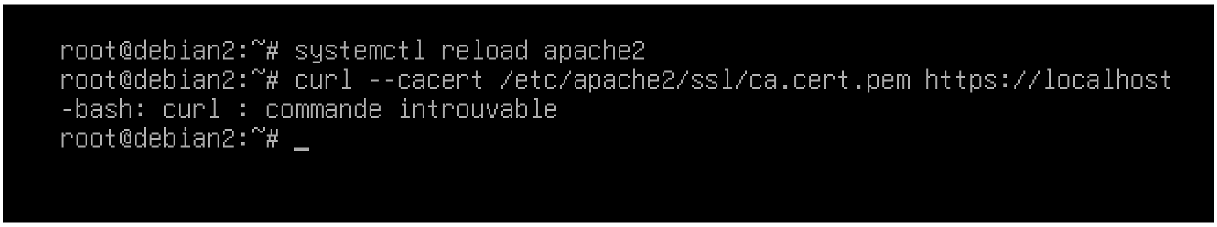


```
root@debian2:~# systemctl reload apache2
root@debian2:~# _
```

Maintenant, testons si HTTPS fonctionne bien sur HTTPS-Server1.

Depuis HTTPS-Server1 (ou depuis un client qui peut atteindre ce serveur), tape :

curl --cacert /etc/apache2/ssl/ca.cert.pem https://localhost

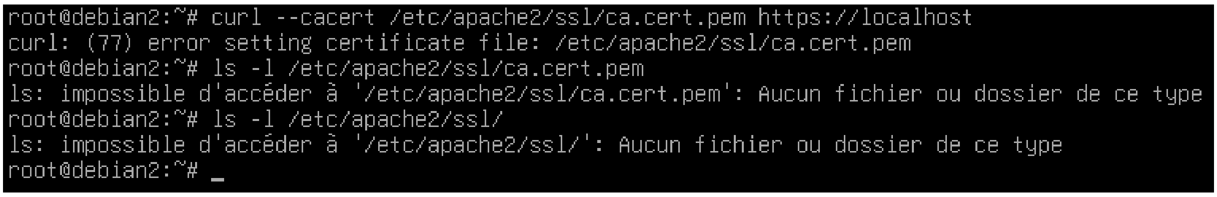


```
root@debian2:~# systemctl reload apache2
root@debian2:~# curl --cacert /etc/apache2/ssl/ca.cert.pem https://localhost
-bash: curl : commande introuvable
root@debian2:~# _
```

La commande curl est introuvable sur debian2, c'est probablement que le paquet n'est pas installé :

apt update

apt install curl



```
root@debian2:~# curl --cacert /etc/apache2/ssl/ca.cert.pem https://localhost
curl: (77) error setting certificate file: /etc/apache2/ssl/ca.cert.pem
root@debian2:~# ls -l /etc/apache2/ssl/ca.cert.pem
ls: impossible d'accéder à '/etc/apache2/ssl/ca.cert.pem': Aucun fichier ou dossier de ce type
root@debian2:~# ls -l /etc/apache2/ssl/
ls: impossible d'accéder à '/etc/apache2/ssl/': Aucun fichier ou dossier de ce type
root@debian2:~# _
```

Ça veut dire que le dossier /etc/apache2/ssl/ n'existe pas encore sur ce serveur.

Donc on va le créer !

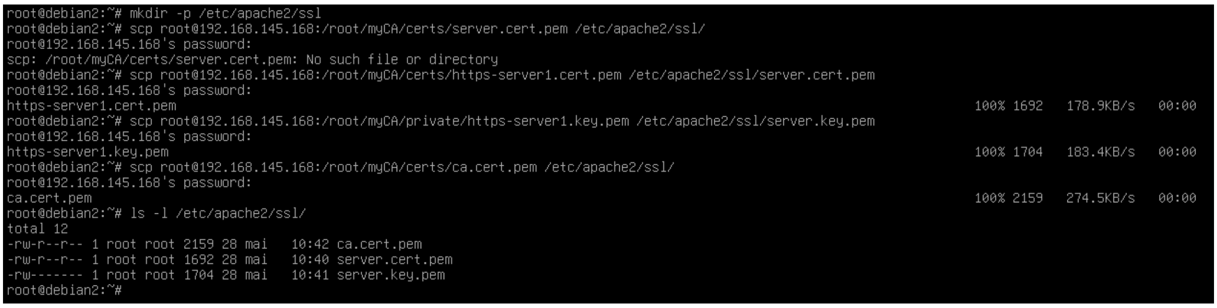
mkdir -p /etc/apache2/ssl

et on exécute ces commandes pour copier les fichiers en les renommant comme Apache attend :

scp root@<IP-debian1>:/root/myCA/certs/https-server1.cert.pem /etc/apache2/ssl/server.cert.pem

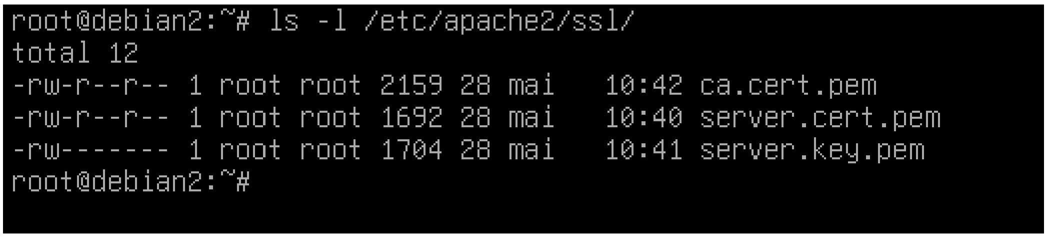
scp root@<IP-debian1>:/root/myCA/private/https-server1.key.pem /etc/apache2/ssl/server.key.pem

scp root@<IP-debian1>:/root/myCA/certs/ca.cert.pem /etc/apache2/ssl/



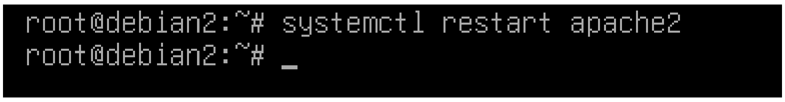
```
root@debian2:~# mkdir -p /etc/apache2/ssl
root@debian2:~# scp root@192.168.145.168:/root/myCA/certs/server.cert.pem /etc/apache2/ssl/
root@192.168.145.168's password:
scp: /root/myCA/certs/server.cert.pem: No such file or directory
root@debian2:~# scp root@192.168.145.168:/root/myCA/certs/https-server1.cert.pem /etc/apache2/ssl/server.cert.pem
root@192.168.145.168's password:
https-server1.cert.pem                                                                  100% 1692   178.9KB/s   00:00
root@debian2:~# scp root@192.168.145.168:/root/myCA/private/https-server1.key.pem /etc/apache2/ssl/server.key.pem
root@192.168.145.168's password:
https-server1.key.pem                                                                   100% 1704   183.4KB/s   00:00
root@debian2:~# scp root@192.168.145.168:/root/myCA/certs/ca.cert.pem /etc/apache2/ssl/
root@192.168.145.168's password:
ca.cert.pem                                                                             100% 2159   274.5KB/s   00:00
root@debian2:~# ls -l /etc/apache2/ssl/
total 12
-rw-r--r-- 1 root root 2159 28 mai   10:42 ca.cert.pem
-rw-r--r-- 1 root root 1692 28 mai   10:40 server.cert.pem
-rw------- 1 root root 1704 28 mai   10:41 server.key.pem
root@debian2:~#
```

Ensuite, vérifie la présence des fichiers :



```
root@debian2:~# ls -l /etc/apache2/ssl/
total 12
-rw-r--r-- 1 root root 2159 28 mai   10:42 ca.cert.pem
-rw-r--r-- 1 root root 1692 28 mai   10:40 server.cert.pem
-rw------- 1 root root 1704 28 mai   10:41 server.key.pem
root@debian2:~#
```

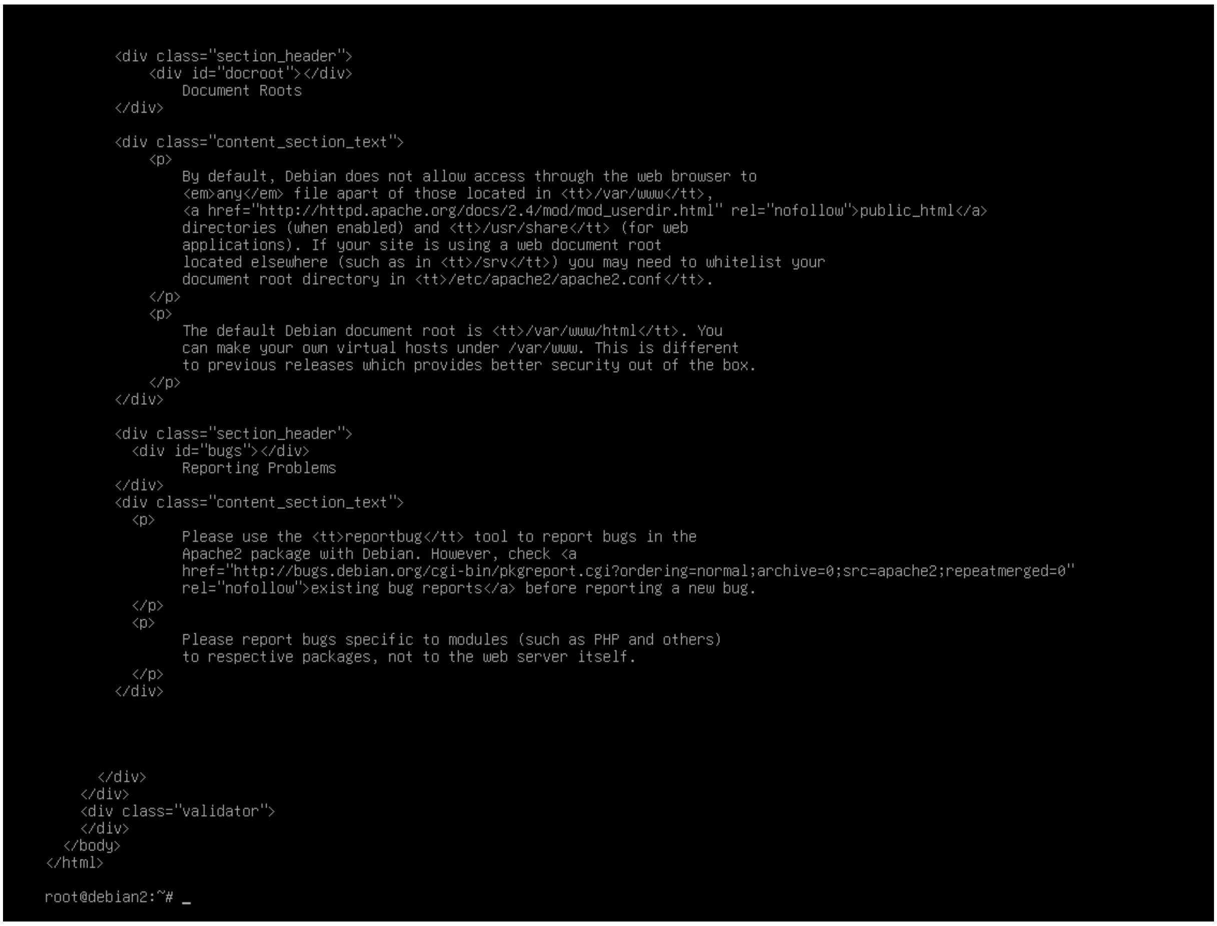
Puis redémarre Apache2 :



```
root@debian2:~# systemctl restart apache2
root@debian2:~# _
```

Une fois ça fait, re-tente :

curl --cacert /etc/apache2/ssl/ca.cert.pem https://192.168.145.170



```
        <div class="section_header">
            <div id="docroot"></div>
                Document Roots
        </div>

        <div class="content_section_text">
            <p>
                By default, Debian does not allow access through the web browser to
                <em>any</em> file apart of those located in <tt>/var/www</tt>,
                <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html" rel="nofollow">public_html</a>
                directories (when enabled) and <tt>/usr/share</tt> (for web
                applications). If your site is using a web document root
                located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
                document root directory in <tt>/etc/apache2/apache2.conf</tt>.
            </p>
            <p>
                The default Debian document root is <tt>/var/www/html</tt>. You
                can make your own virtual hosts under /var/www. This is different
                to previous releases which provides better security out of the box.
            </p>
        </div>

        <div class="section_header">
          <div id="bugs"></div>
                Reporting Problems
        </div>
        <div class="content_section_text">
          <p>
                Please use the <tt>reportbug</tt> tool to report bugs in the
                Apache2 package with Debian. However, check <a
                href="http://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apache2;repeatmerged=0"
                rel="nofollow">existing bug reports</a> before reporting a new bug.
          </p>
          <p>
                Please report bugs specific to modules (such as PHP and others)
                to respective packages, not to the web server itself.
          </p>
        </div>




      </div>
    </div>
    <div class="validator">
    </div>
  </body>
</html>

root@debian2:~# _
```

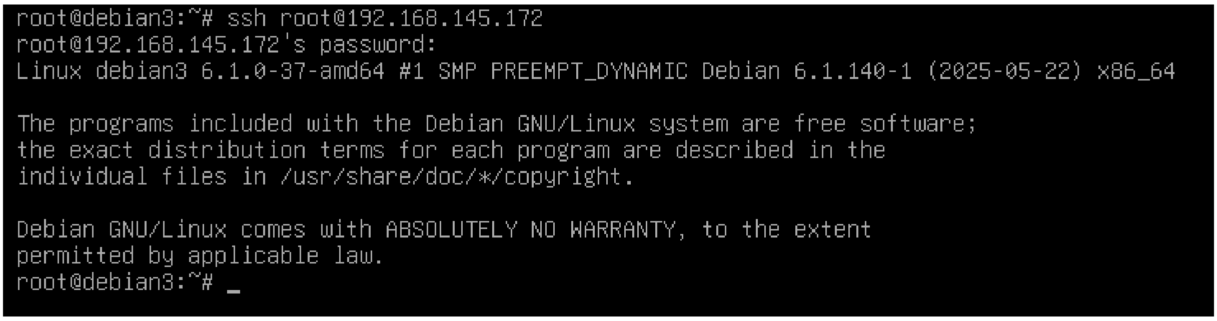
Le certificat est bien utilisé.

curl a fait une requête HTTPS **vérifiée avec le certificat de ta CA**.

Le serveur Apache **répond avec une page HTML** (la page par défaut ou celle configurée).

Connection sur le serveur HTTPS-Server2

ssh root@192.168.145.172



```
root@debian3:~# ssh root@192.168.145.172
root@192.168.145.172's password:
Linux debian3 6.1.0-37-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.140-1 (2025-05-22) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian3:~# _
```

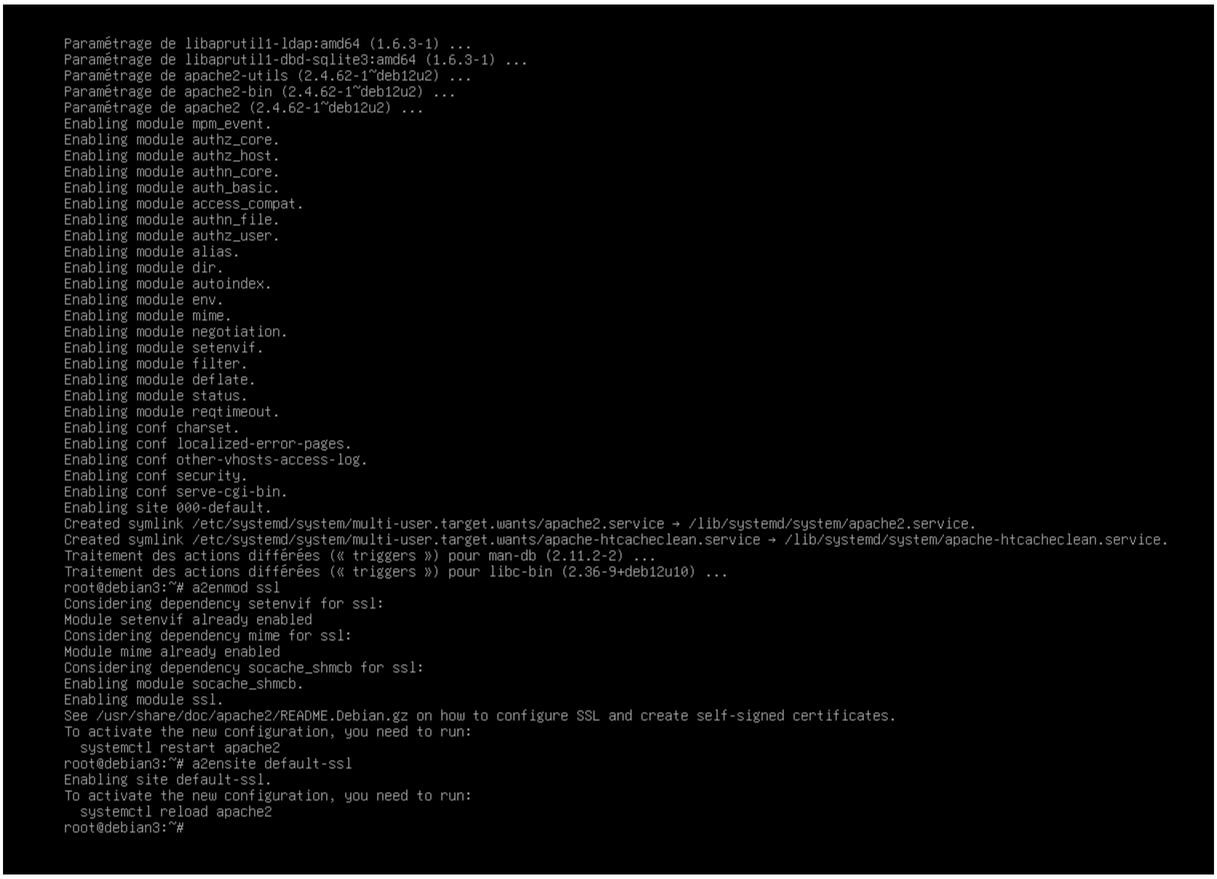
Installe Apache2

apt update

apt install apache2 -y

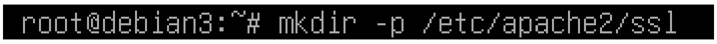
a2enmod ssl

a2ensite default-ssl



```
Paramétrage de libaprutil1-ldap:amd64 (1.6.3-1) ...
Paramétrage de libaprutil1-dbd-sqlite3:amd64 (1.6.3-1) ...
Paramétrage de apache2-utils (2.4.62-1~deb12u2) ...
Paramétrage de apache2-bin (2.4.62-1~deb12u2) ...
Paramétrage de apache2 (2.4.62-1~deb12u2) ...
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.36-9+deb12u10) ...
root@debian3:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian3:~# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian3:~#
```

Crée le dossier SSL



```
root@debian3:~# mkdir -p /etc/apache2/ssl
```

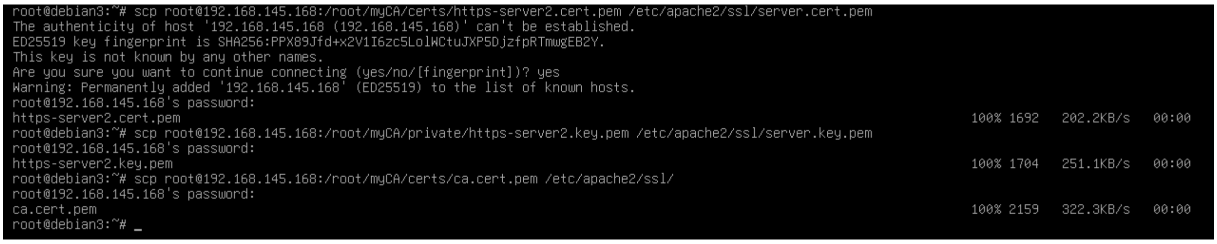
Copie les fichiers depuis le serveur CA

Depuis HTTPS-Server2 faire :

scp root@192.168.145.168:/root/myCA/certs/https-server2.cert.pem /etc/apache2/ssl/server.cert.pem

scp root@192.168.145.168:/root/myCA/private/https-server2.key.pem /etc/apache2/ssl/server.key.pem

scp root@192.168.145.168:/root/myCA/certs/ca.cert.pem /etc/apache2/ssl/



```
root@debian3:~# scp root@192.168.145.168:/root/myCA/certs/https-server2.cert.pem /etc/apache2/ssl/server.cert.pem
The authenticity of host '192.168.145.168 (192.168.145.168)' can't be established.
ED25519 key fingerprint is SHA256:PPX89Jfd+x2V1I6zc5LolWCtuJXP5DjzfpRTmwgEB2Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.145.168' (ED25519) to the list of known hosts.
root@192.168.145.168's password:
https-server2.cert.pem                                                                   100% 1692   202.2KB/s   00:00
root@debian3:~# scp root@192.168.145.168:/root/myCA/private/https-server2.key.pem /etc/apache2/ssl/server.key.pem
root@192.168.145.168's password:
https-server2.key.pem                                                                    100% 1704   251.1KB/s   00:00
root@debian3:~# scp root@192.168.145.168:/root/myCA/certs/ca.cert.pem /etc/apache2/ssl/
root@192.168.145.168's password:
ca.cert.pem                                                                              100% 2159   322.3KB/s   00:00
root@debian3:~# _
```

Configure Apache pour utiliser HTTPS

nano /etc/apache2/sites-available/default-ssl.conf

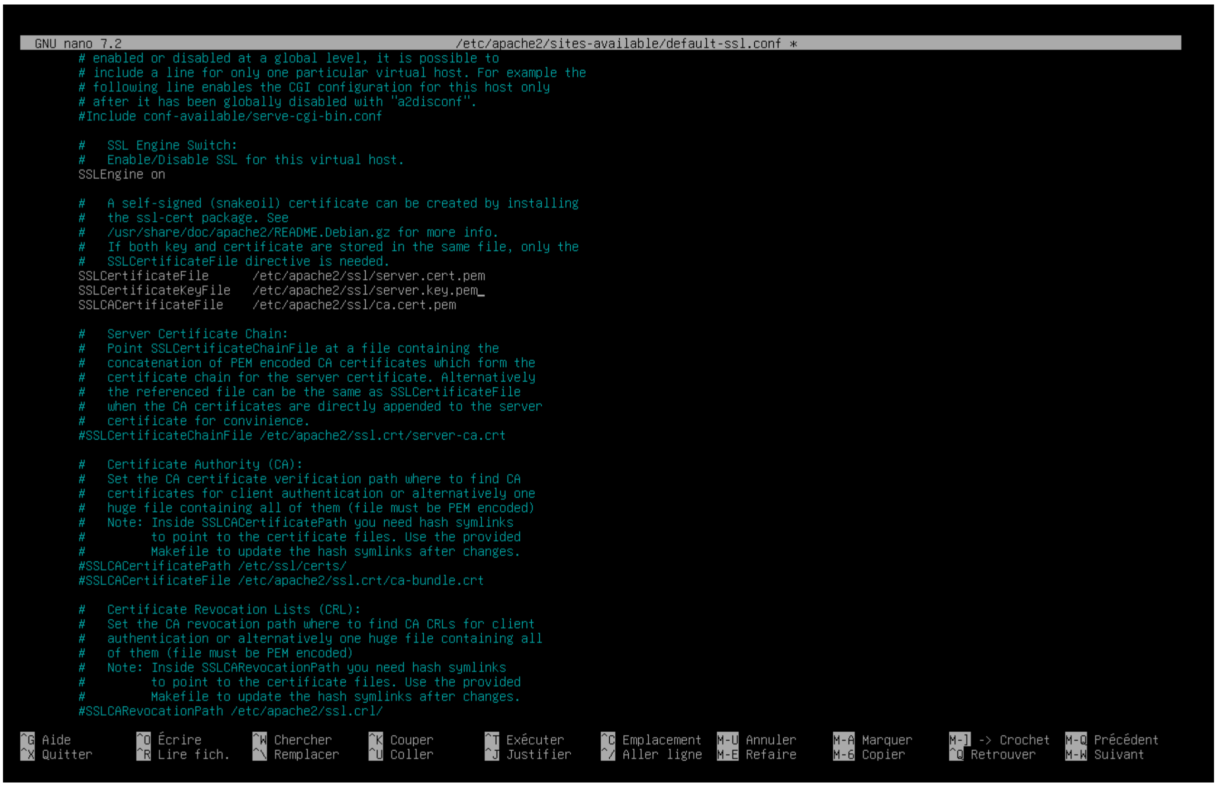
Assure-toi que tu as bien ces lignes dans la section <VirtualHost _default_:443> :

SSLEngine on

SSLCertificateFile   /etc/apache2/ssl/server.cert.pem

SSLCertificateKeyFile   /etc/apache2/ssl/server.key.pem

SSLCACertificateFile   /etc/apache2/ssl/ca.cert.pem



```
  GNU nano 7.2                                       /etc/apache2/sites-available/default-ssl.conf *
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/ssl/server.cert.pem
        SSLCertificateKeyFile   /etc/apache2/ssl/server.key.pem_
        SSLCACertificateFile    /etc/apache2/ssl/ca.cert.pem

        #   Server Certificate Chain:
        #   Point SSLCertificateChainFile at a file containing the
        #   concatenation of PEM encoded CA certificates which form the
        #   certificate chain for the server certificate. Alternatively
        #   the referenced file can be the same as SSLCertificateFile
        #   when the CA certificates are directly appended to the server
        #   certificate for convinience.
        #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

        #   Certificate Authority (CA):
        #   Set the CA certificate verification path where to find CA
        #   certificates for client authentication or alternatively one
        #   huge file containing all of them (file must be PEM encoded)
        #   Note: Inside SSLCACertificatePath you need hash symlinks
        #         to point to the certificate files. Use the provided
        #         Makefile to update the hash symlinks after changes.
        #SSLCACertificatePath /etc/ssl/certs/
        #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

        #   Certificate Revocation Lists (CRL):
        #   Set the CA revocation path where to find CA CRLs for client
        #   authentication or alternatively one huge file containing all
        #   of them (file must be PEM encoded)
        #   Note: Inside SSLCARevocationPath you need hash symlinks
        #         to point to the certificate files. Use the provided
        #         Makefile to update the hash symlinks after changes.
        #SSLCARevocationPath /etc/apache2/ssl.crl/

^G Aide       ^O Écrire     ^W Chercher   ^K Couper     ^T Exécuter   ^C Emplacement M-U Annuler   M-A Marquer   M-] -> Crochet M-Q Précédent
^X Quitter    ^R Lire fich. ^\ Remplacer  ^U Coller     ^J Justifier  ^/ Aller ligne M-E Refaire   M-6 Copier    ^Q Retrouver   M-W Suivant
```

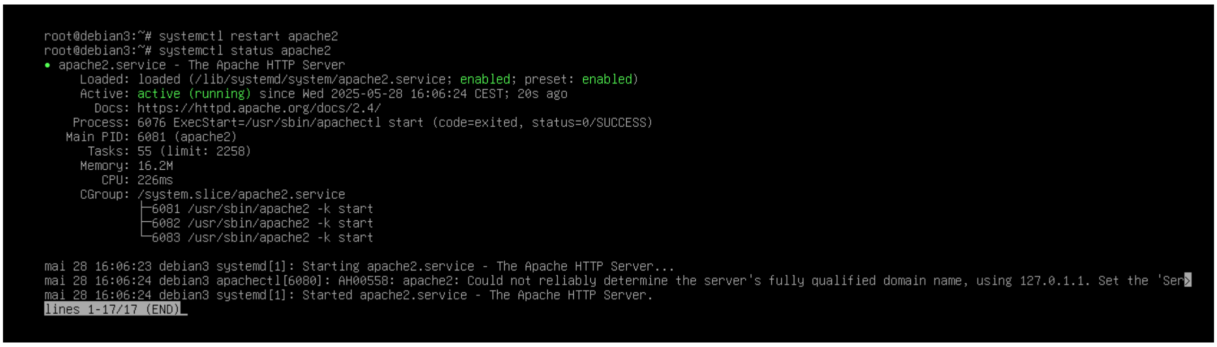
Si ça existe déjà, modifie seulement les chemins.

Redémarre Apache

systemctl restart apache2

Vérifie ensuite :

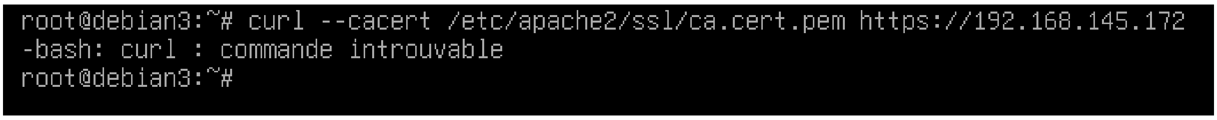
systemctl status apache2



```
root@debian3:~# systemctl restart apache2
root@debian3:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-05-28 16:06:24 CEST; 20s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 6076 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 6081 (apache2)
      Tasks: 55 (limit: 2258)
     Memory: 16.2M
        CPU: 226ms
     CGroup: /system.slice/apache2.service
             ├─6081 /usr/sbin/apache2 -k start
             ├─6082 /usr/sbin/apache2 -k start
             └─6083 /usr/sbin/apache2 -k start

mai 28 16:06:23 debian3 systemd[1]: Starting apache2.service - The Apache HTTP Server...
mai 28 16:06:24 debian3 apachectl[6080]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'Ser>
mai 28 16:06:24 debian3 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-17/17 (END)_
```

Teste avec curl

curl --cacert /etc/apache2/ssl/ca.cert.pem https://192.168.145.172

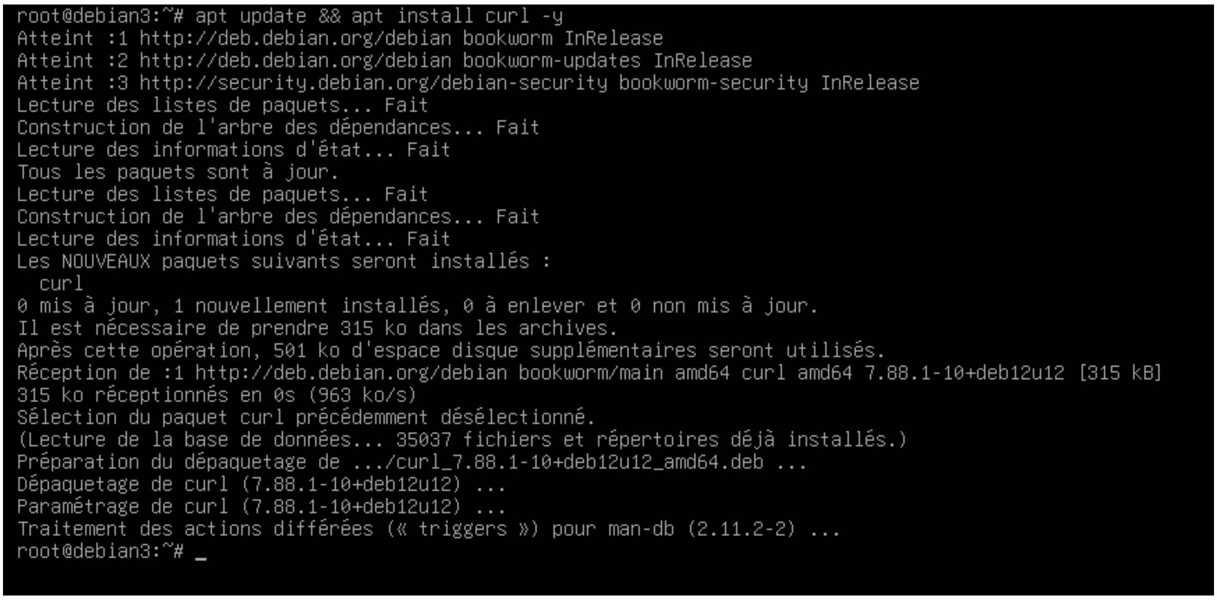


```
root@debian3:~# curl --cacert /etc/apache2/ssl/ca.cert.pem https://192.168.145.172
-bash: curl : commande introuvable
root@debian3:~#
```

Pas de souci, c'est simplement que curl n'est pas encore installé sur HTTPS-Server2.

Voici la commande pour l'installer :

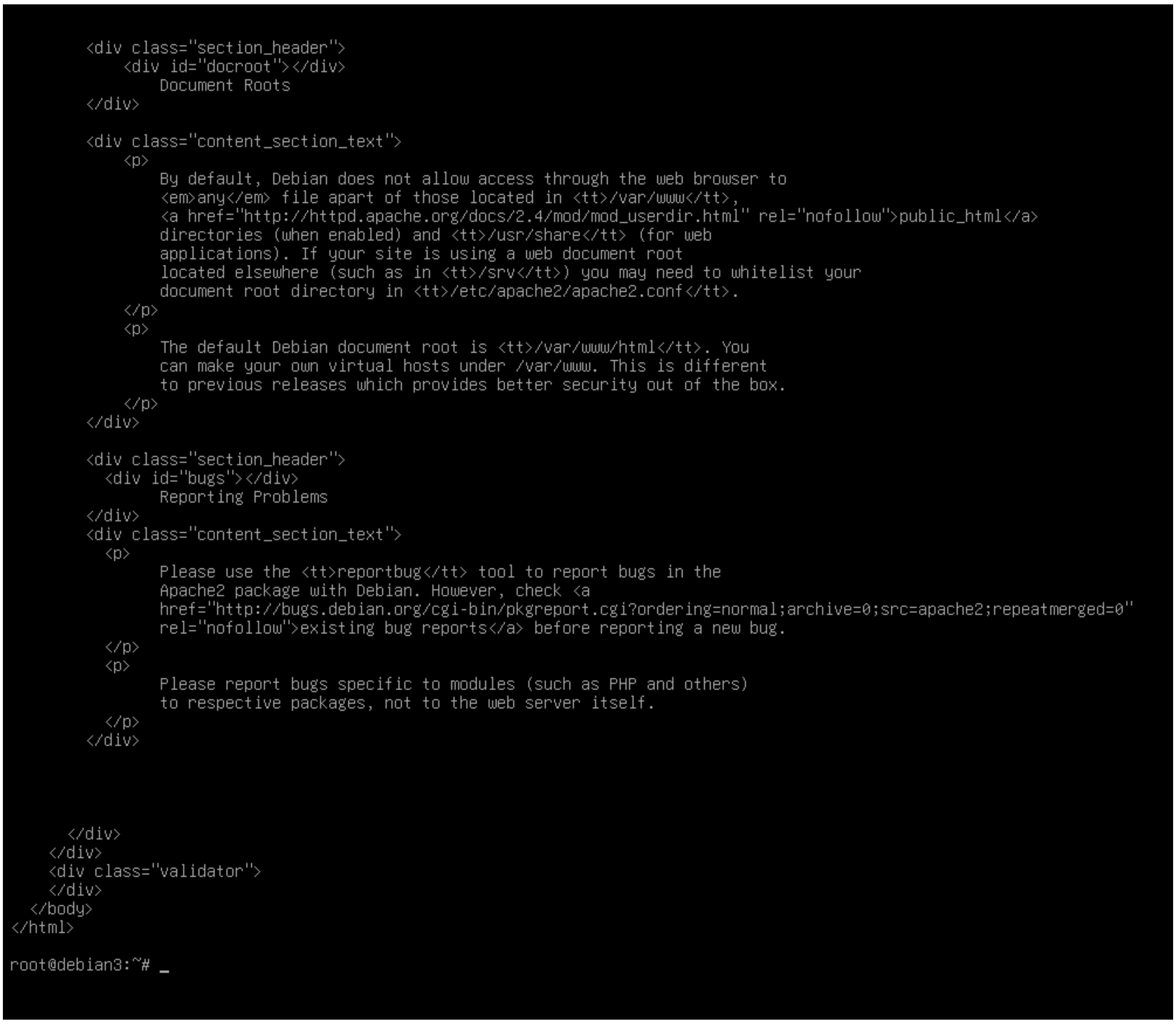
apt update && apt install curl -y



```
root@debian3:~# apt update && apt install curl -y
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://deb.debian.org/debian bookworm-updates InRelease
Atteint :3 http://security.debian.org/debian-security bookworm-security InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  curl
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 315 ko dans les archives.
Après cette opération, 501 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://deb.debian.org/debian bookworm/main amd64 curl amd64 7.88.1-10+deb12u12 [315 kB]
315 ko réceptionnés en 0s (963 ko/s)
Sélection du paquet curl précédemment désélectionné.
(Lecture de la base de données... 35037 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../curl_7.88.1-10+deb12u12_amd64.deb ...
Dépaquetage de curl (7.88.1-10+deb12u12) ...
Paramétrage de curl (7.88.1-10+deb12u12) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
root@debian3:~# _
```

Une fois installer, relancer:

curl --cacert /etc/apache2/ssl/ca.cert.pem https://192.168.145.172



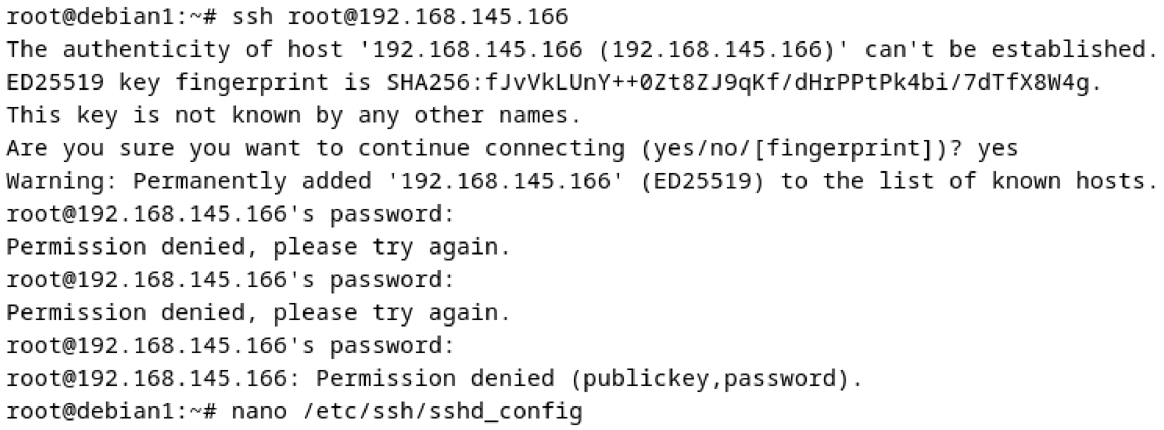
```
        <div class="section_header">
            <div id="docroot"></div>
                Document Roots
        </div>

        <div class="content_section_text">
            <p>
                By default, Debian does not allow access through the web browser to
                <em>any</em> file apart of those located in <tt>/var/www</tt>,
                <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html" rel="nofollow">public_html</a>
                directories (when enabled) and <tt>/usr/share</tt> (for web
                applications). If your site is using a web document root
                located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
                document root directory in <tt>/etc/apache2/apache2.conf</tt>.
            </p>
            <p>
                The default Debian document root is <tt>/var/www/html</tt>. You
                can make your own virtual hosts under /var/www. This is different
                to previous releases which provides better security out of the box.
            </p>
        </div>

        <div class="section_header">
          <div id="bugs"></div>
                Reporting Problems
        </div>
        <div class="content_section_text">
          <p>
                Please use the <tt>reportbug</tt> tool to report bugs in the
                Apache2 package with Debian. However, check <a
                href="http://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apache2;repeatmerged=0"
                rel="nofollow">existing bug reports</a> before reporting a new bug.
          </p>
          <p>
                Please report bugs specific to modules (such as PHP and others)
                to respective packages, not to the web server itself.
          </p>
        </div>




      </div>
    </div>
    <div class="validator">
    </div>
  </body>
</html>

root@debian3:~# _
```

Le certificat est bien utilisé.

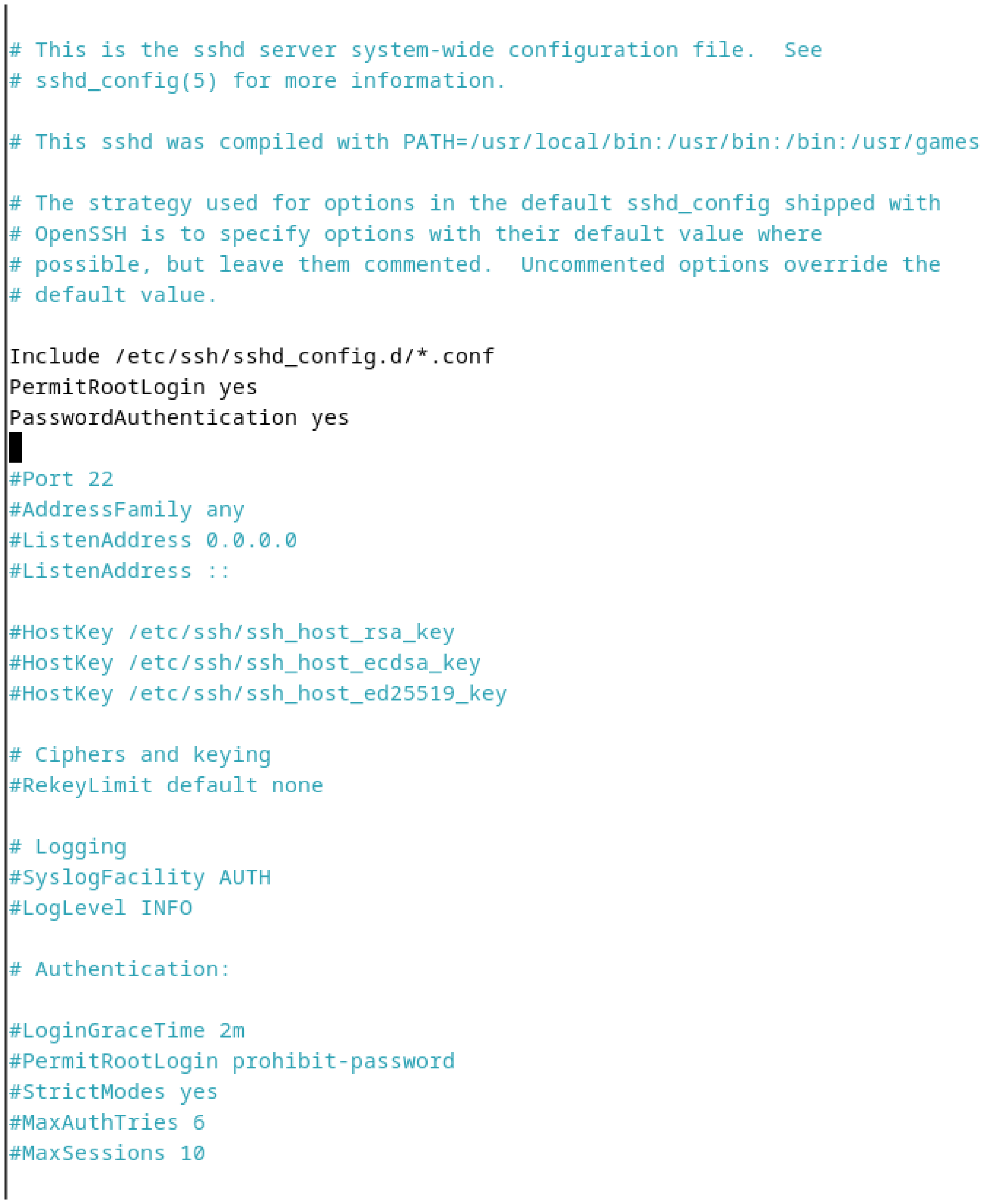
curl a fait une requête HTTPS **vérifiée avec le certificat de ta CA**.

Le serveur Apache **répond avec une page HTML** (la page par défaut ou celle configurée).

Test de la connexion HTTPS depuis les clients

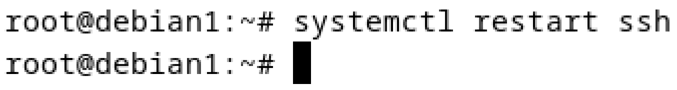


```
root@debian1:~# ssh root@192.168.145.166
The authenticity of host '192.168.145.166 (192.168.145.166)' can't be established.
ED25519 key fingerprint is SHA256:fJvVkLUnY++0Zt8ZJ9qKf/dHrPPtPk4bi/7dTfX8W4g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.145.166' (ED25519) to the list of known hosts.
root@192.168.145.166's password:
Permission denied, please try again.
root@192.168.145.166's password:
Permission denied, please try again.
root@192.168.145.166's password:
root@192.168.145.166: Permission denied (publickey,password).
root@debian1:~# nano /etc/ssh/sshd_config
```



```

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf
PermitRootLogin yes
PasswordAuthentication yes

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

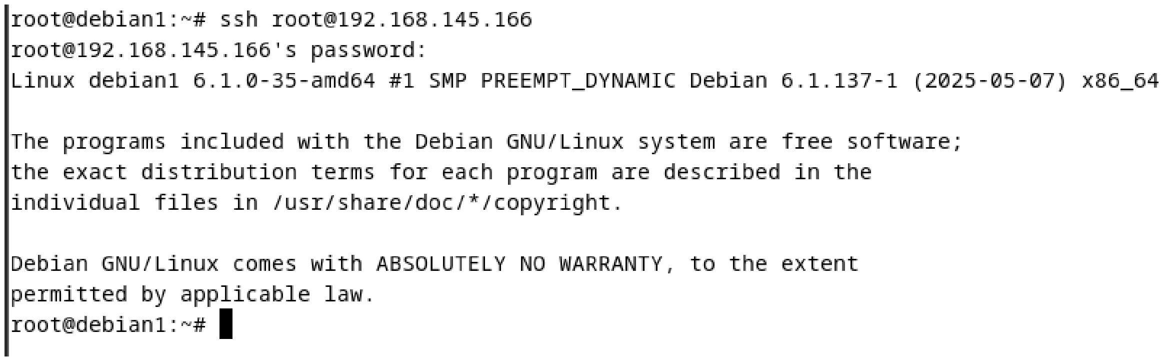
Ensuite, redémarre le service SSH :



```
root@debian1:~# systemctl restart ssh
root@debian1:~# 
```

Teste à nouveau la connexion

ssh root@192.168.145.171



```
root@debian1:~# ssh root@192.168.145.166
root@192.168.145.166's password:
Linux debian1 6.1.0-35-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.137-1 (2025-05-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian1:~# 
```

Tester la connexion HTTPS avec curl

Mais avant, installe curl s'il n'est pas déjà là :

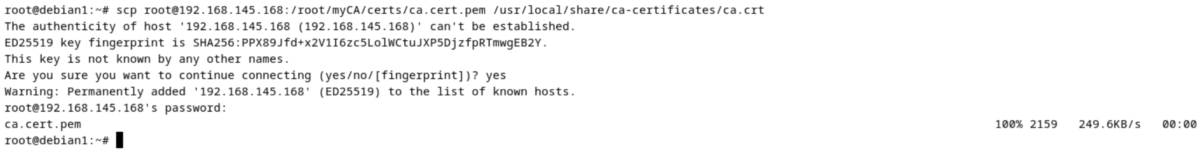
apt update && apt install curl -y

Ensuite, teste :

curl --cacert /etc/ssl/certs/ca.cert.pem https://192.168.145.170

Si tu es sur la machine CA-Server et que tu veux copier vers ta machine client, tu peux utiliser la commande scp depuis la machine client comme ceci :

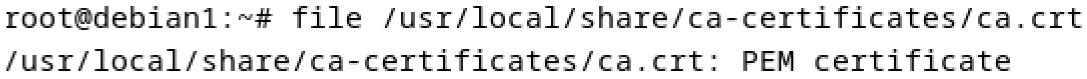
scp root@192.168.145.168:/root/myCA/certs/ca.cert.pem /usr/local/share/ca-certificates/ca.crt



```
root@debian1:~# scp root@192.168.145.168:/root/myCA/certs/ca.cert.pem /usr/local/share/ca-certificates/ca.crt
The authenticity of host '192.168.145.168 (192.168.145.168)' can't be established.
ED25519 key fingerprint is SHA256:PPX89Jfd+x2V1I6zc5LolWCtuJXP5DjzfpRTmwgEB2Y.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.145.168' (ED25519) to the list of known hosts.
root@192.168.145.168's password:
ca.cert.pem                                                                                                                   100% 2159   249.6KB/s   00:00
root@debian1:~# 
```

1. **Vérifie le fichier copié :**

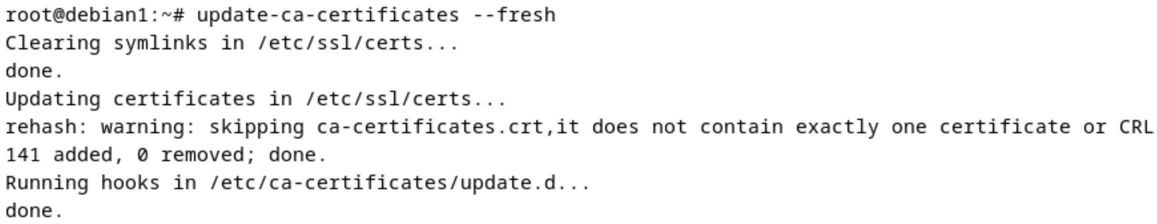
file /usr/local/share/ca-certificates/ca.crt



```
root@debian1:~# file /usr/local/share/ca-certificates/ca.crt
/usr/local/share/ca-certificates/ca.crt: PEM certificate
```

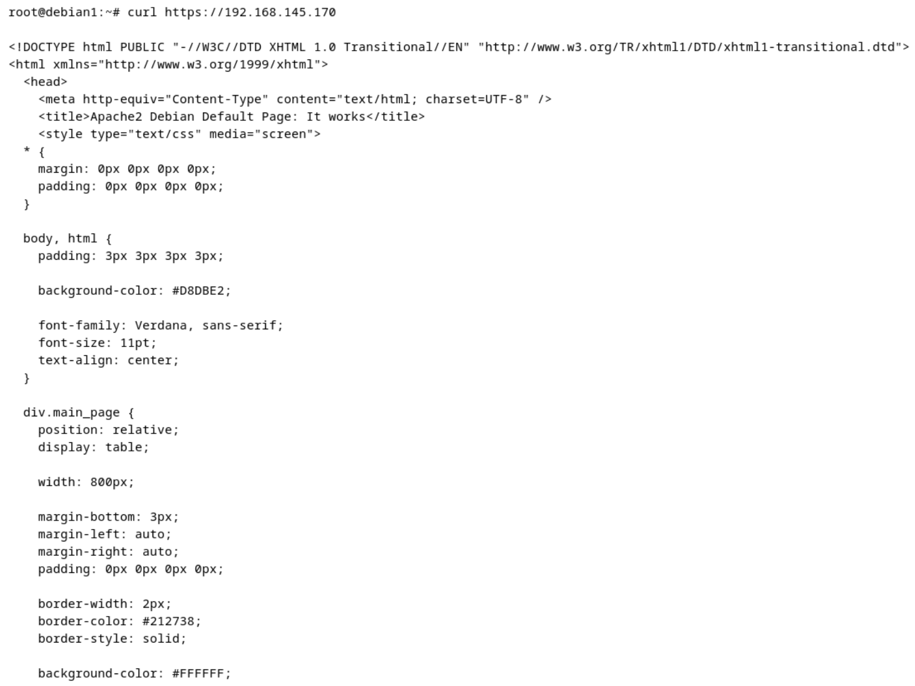
Maintenant, essayons la mise à jour forcée du magasin de certificats :

Update-ca-certificates –fresh



```
root@debian1:~# update-ca-certificates --fresh
Clearing symlinks in /etc/ssl/certs...
done.
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
141 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

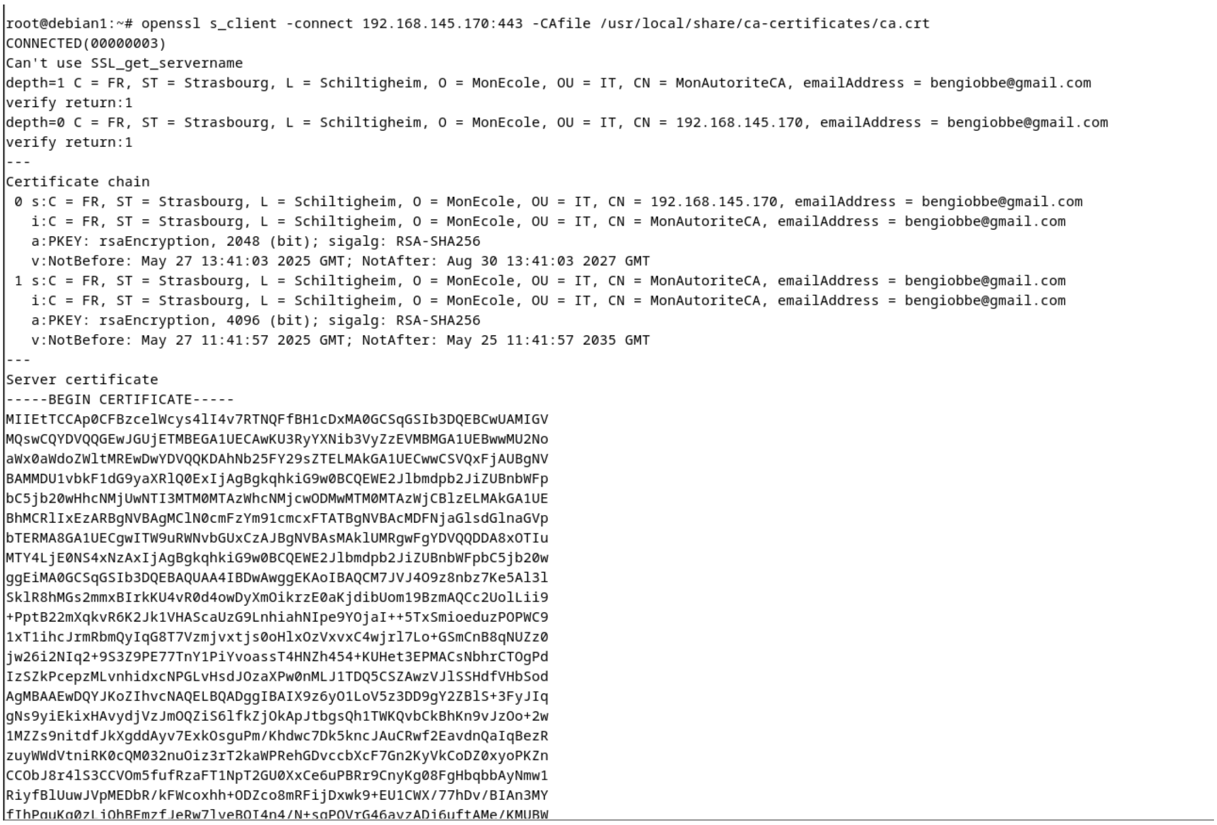
Après on, retente le test HTTPS avec curl : curl https://192.168.145.170



```
root@debian1:~# curl https://192.168.145.170

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Debian Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
  }

  div.main_page {
    position: relative;
    display: table;

    width: 800px;

    margin-bottom: 3px;
    margin-left: auto;
    margin-right: auto;
    padding: 0px 0px 0px 0px;

    border-width: 2px;
    border-color: #212738;
    border-style: solid;

    background-color: #FFFFFF;
```

Pour vérifier que c'est bien sécurisé avec ton certificat CA, tu peux faire un test plus précis avec openssl :

openssl s_client -connect 192.168.145.170:443 -CAfile /usr/local/share/ca-certificates/ca.crt



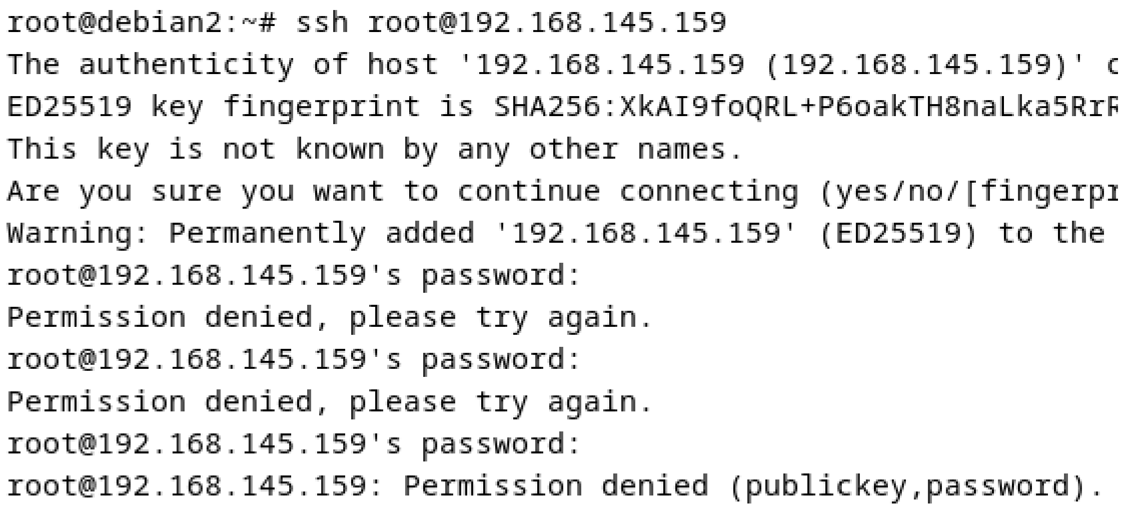
```
root@debian1:~# openssl s_client -connect 192.168.145.170:443 -CAfile /usr/local/share/ca-certificates/ca.crt
CONNECTED(00000003)
Can't use SSL_get_servername
depth=1 C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = MonAutoriteCA, emailAddress = bengiobbe@gmail.com
verify return:1
depth=0 C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = 192.168.145.170, emailAddress = bengiobbe@gmail.com
verify return:1
---
Certificate chain
 0 s:C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = 192.168.145.170, emailAddress = bengiobbe@gmail.com
   i:C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = MonAutoriteCA, emailAddress = bengiobbe@gmail.com
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: May 27 13:41:03 2025 GMT; NotAfter: Aug 30 13:41:03 2027 GMT
 1 s:C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = MonAutoriteCA, emailAddress = bengiobbe@gmail.com
   i:C = FR, ST = Strasbourg, L = Schiltigheim, O = MonEcole, OU = IT, CN = MonAutoriteCA, emailAddress = bengiobbe@gmail.com
   a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
   v:NotBefore: May 27 11:41:57 2025 GMT; NotAfter: May 25 11:41:57 2035 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEtTCCAp0CFBzcelWcys4lI4v7RTNQFfBH1cDxMA0GCSqGSIb3DQEBCwUAMIGV
MQswCQYDVQQGEwJGUjETMBEGA1UECAwKU3RyYXNib3VyZzEVMBMGA1UEBwwMU2No
aWx0aWdoZWltMREwDwYDVQQKDAhNb25FY29sZTELMAkGA1UECwwCSVQxFjAUBgNV
BAMMDU1vbkF1dG9yaXRlQ0ExIjAgBgkqhkiG9w0BCQEWE2Jlbmdpb2JiZUBnbWFp
bC5jb20wHhcNMjUwNTI3MTM0MTAzWhcNMjcwODMwMTM0MTAzWjCBlzELMAkGA1UE
BhMCRlIxEzARBgNVBAgMClN0cmFzYm91cmcxFTATBgNVBAcMDFNjaGlsdGlnaGVp
bTERMA8GA1UECgwITW9uRWNvbGUxCzAJBgNVBAsMAklUMRgwFgYDVQQDDA8xOTIu
MTY4LjE0NS4xNzAxIjAgBgkqhkiG9w0BCQEWE2Jlbmdpb2JiZUBnbWFpbC5jb20w
ggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCM7JVJ4O9z8nbz7Ke5Al3l
SklR8hMGs2mmxBIrkKU4vR0d4owDyXmOikrzE0aKjdibUom19BzmAQCc2UolLii9
+PptB22mXqkvR6K2Jk1VHAScaUzG9LnhiahNIpe9YOjaI++5TxSmioeduzPOPWC9
1xT1ihcJrmRbmQyIqG8T7Vzmjvxtjs0oHlxOzVxvxC4wjrl7Lo+GSmCnB8qNUZz0
jw26i2NIq2+9S3Z9PE77TnY1PiYvoassT4HNZh454+KUHet3EPMACsNbhrCTOgPd
IzSZkPcepzMLvnhidxcNPGLvHsdJOzaXPw0nMLJ1TDQ5CSZAwzVJlSSHdfVHbSod
AgMBAAEwDQYJKoZIhvcNAQELBQADggIBAIX9z6yO1LoV5z3DD9gY2ZBlS+3FyJIq
gNs9yiEkixHAvydjVzJmOQZiS6lfkZjOkApJtbgsQh1TWKQvbCkBhKn9vJzOo+2w
1MZZs9nitdfJkXgddAyv7ExkOsguPm/Khdwc7Dk5kncJAuCRwf2EavdnQaIqBezR
zuyWWdVtniRK0cQM032nuOiz3rT2kaWPRehGDvccbXcF7Gn2KyVkCoDZ0xyoPKZn
CCObJ8r4lS3CCVOm5fufRzaFT1NpT2GU0XxCe6uPBRr9CnyKg08FgHbqbbAyNmw1
RiyfBlUuwJVpMEDbR/kFWcoxhh+ODZco8mRFijDxwk9+EU1CWX/77hDv/BIAn3MY
fIhPguKg0zLjOhBEmzfJeRw7lyeBOI4n4/N+sgPOVrG46ayzADj6uftAMe/KMUBW
```

Conclusion :

Le serveur HTTPS1 fonctionne correctement avec ton certificat signé par ta propre autorité. Tu as bien installé la CA (ca.cert.pem) sur le client, et elle est utilisée avec succès.

Dabord se connecter en SSH à Client2

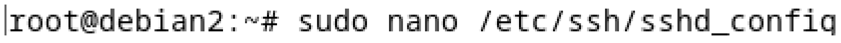
ssh root@192.168.145.159



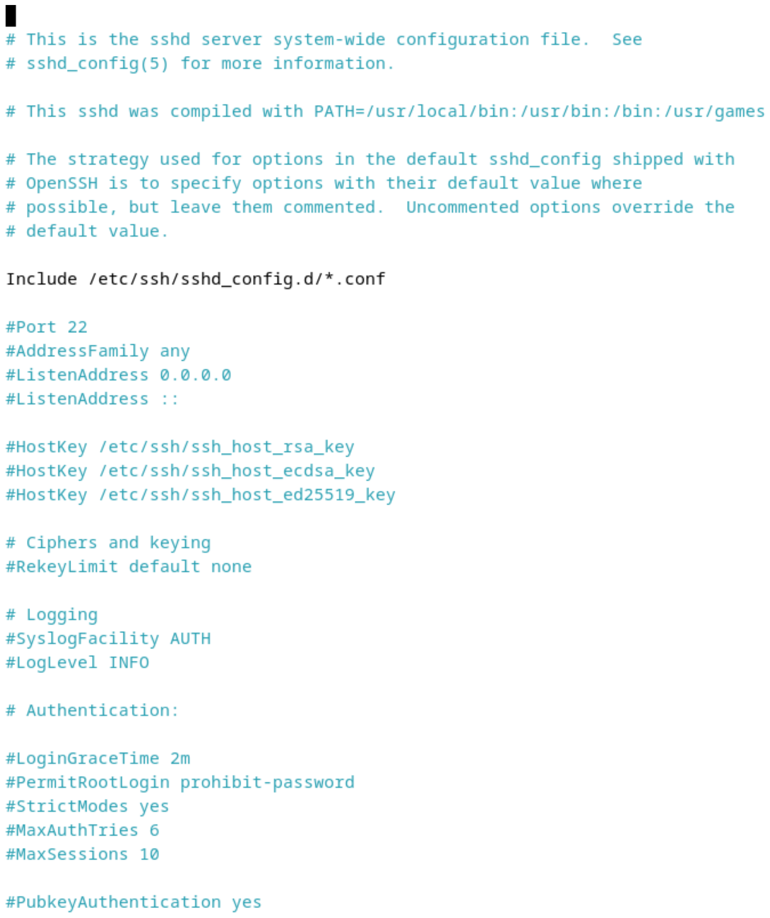
```
root@debian2:~# ssh root@192.168.145.159
The authenticity of host '192.168.145.159 (192.168.145.159)' c
ED25519 key fingerprint is SHA256:XkAI9foQRL+P6oakTH8naLka5RrR
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerpr
Warning: Permanently added '192.168.145.159' (ED25519) to the
root@192.168.145.159's password:
Permission denied, please try again.
root@192.168.145.159's password:
Permission denied, please try again.
root@192.168.145.159's password:
root@192.168.145.159: Permission denied (publickey,password).
```

1. **Édite le fichier sshd_config :**

sudo nano /etc/ssh/sshd_config



```
root@debian2:~# sudo nano /etc/ssh/sshd_config
```



```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes
```

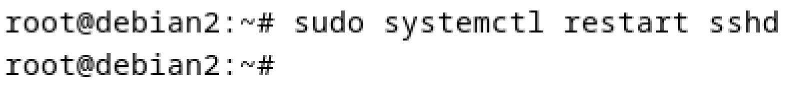
Modifie ou ajoute les lignes suivantes :

PermitRootLogin yes

PasswordAuthentication yes

Enregistre et ferme le fichier (Ctrl+O, Entrée, puis Ctrl+X dans nano).

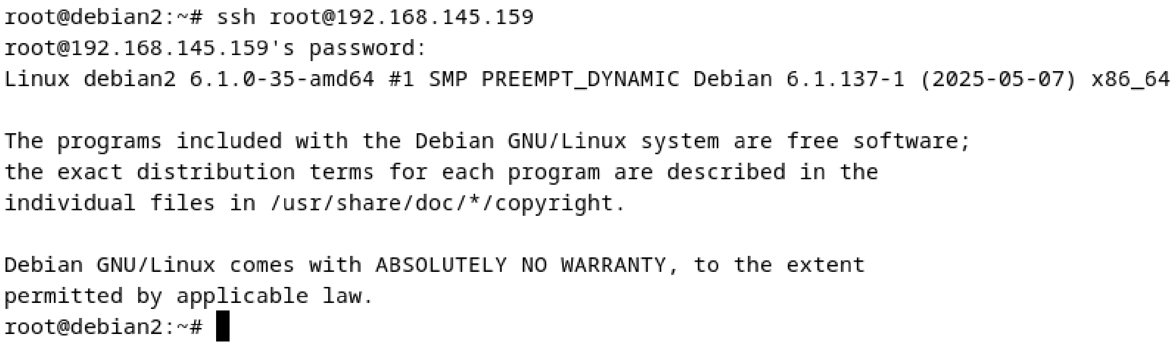
Redémarre le service SSH pour appliquer les changements :



```
root@debian2:~# sudo systemctl restart sshd
root@debian2:~#
```

Refaire :

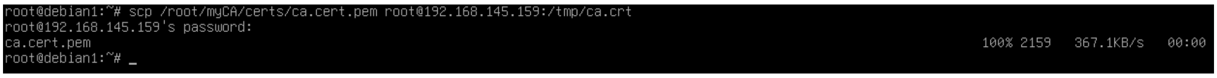
ssh root@192.168.145.159



```
root@debian2:~# ssh root@192.168.145.159
root@192.168.145.159's password:
Linux debian2 6.1.0-35-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.137-1 (2025-05-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian2:~#
```

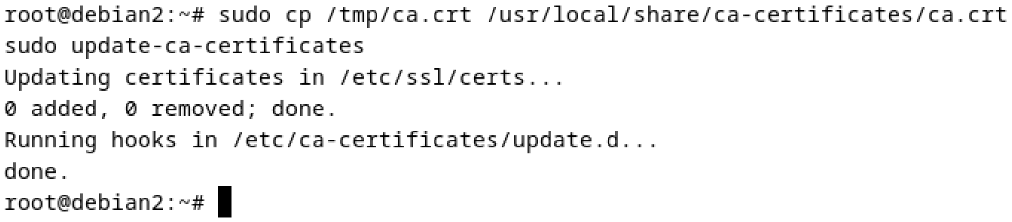
Copier le certificat CA depuis la machine CA vers Client2

scp /root/myCA/certs/ca.cert.pem root@192.168.145.159:/tmp/ca.crt



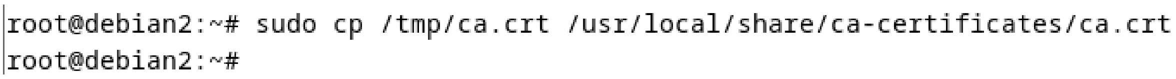
```
root@debian1:~# scp /root/myCA/certs/ca.cert.pem root@192.168.145.159:/tmp/ca.crt
root@192.168.145.159's password:
ca.cert.pem                                                                    100% 2159   367.1KB/s   00:00
root@debian1:~# _
```

Étapes suivantes pour **installer le certificat CA sur le client** :



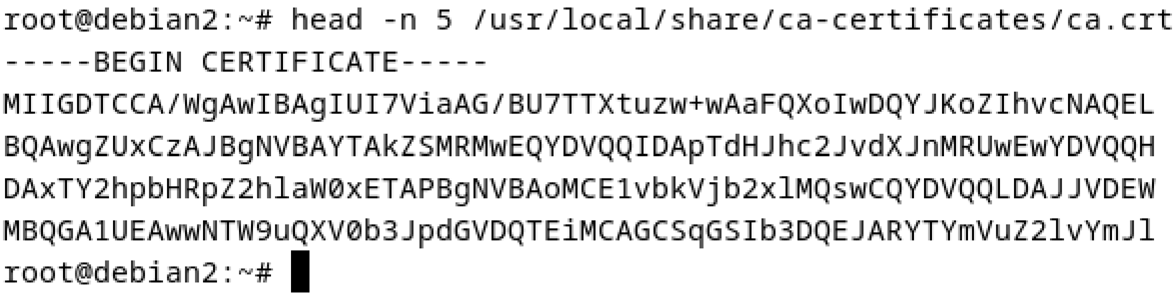
```
root@debian2:~# sudo cp /tmp/ca.crt /usr/local/share/ca-certificates/ca.crt
sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian2:~# █
```

Renomme correctement le fichier avec l’extension .crt si ce n’est pas déjà fait :



```
root@debian2:~# sudo cp /tmp/ca.crt /usr/local/share/ca-certificates/ca.crt
root@debian2:~#
```

Vérifie que le contenu est bien au format PEM :



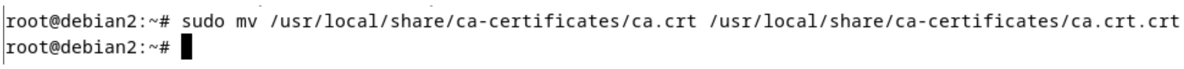
```
root@debian2:~# head -n 5 /usr/local/share/ca-certificates/ca.crt
-----BEGIN CERTIFICATE-----
MIIGDTCCA/WgAwIBAgIUI7ViaAG/BU7TTXtuzw+wAaFQXoIwDQYJKoZIhvcNAQEL
BQAwgZUxCzAJBgNVBAYTAkZSMRMwEQYDVQQIDApTdHJhc2JvdXJnMRUwEwYDVQQH
DAxTY2hpbHRpZ2hlaW0xETAPBgNVBAoMCE1vbkVjb2xlMQswCQYDVQQLDAJJVDEW
MBQGA1UEAwwNTW9uQXV0b3JpdGVDQTEiMCAGCSqGSIb3DQEJARYTYmVuZ2lvYmJl
root@debian2:~# █
```

Maintenant, pourquoi update-ca-certificates n’a pas ajouté ce certificat ?

Il y a probablement une **erreur de nom de fichier ou d’extension**.

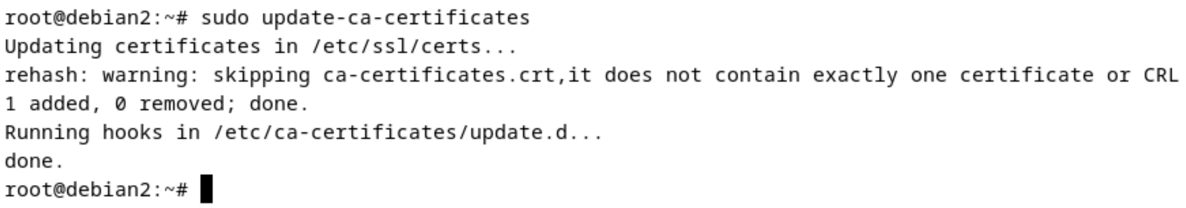
Étapes pour forcer la prise en compte :

Renomme le fichier avec l’extension .crt explicite :



```
root@debian2:~# sudo mv /usr/local/share/ca-certificates/ca.crt /usr/local/share/ca-certificates/ca.crt.crt
root@debian2:~# █
```

Relance la commande :



```
root@debian2:~# sudo update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@debian2:~# █
```

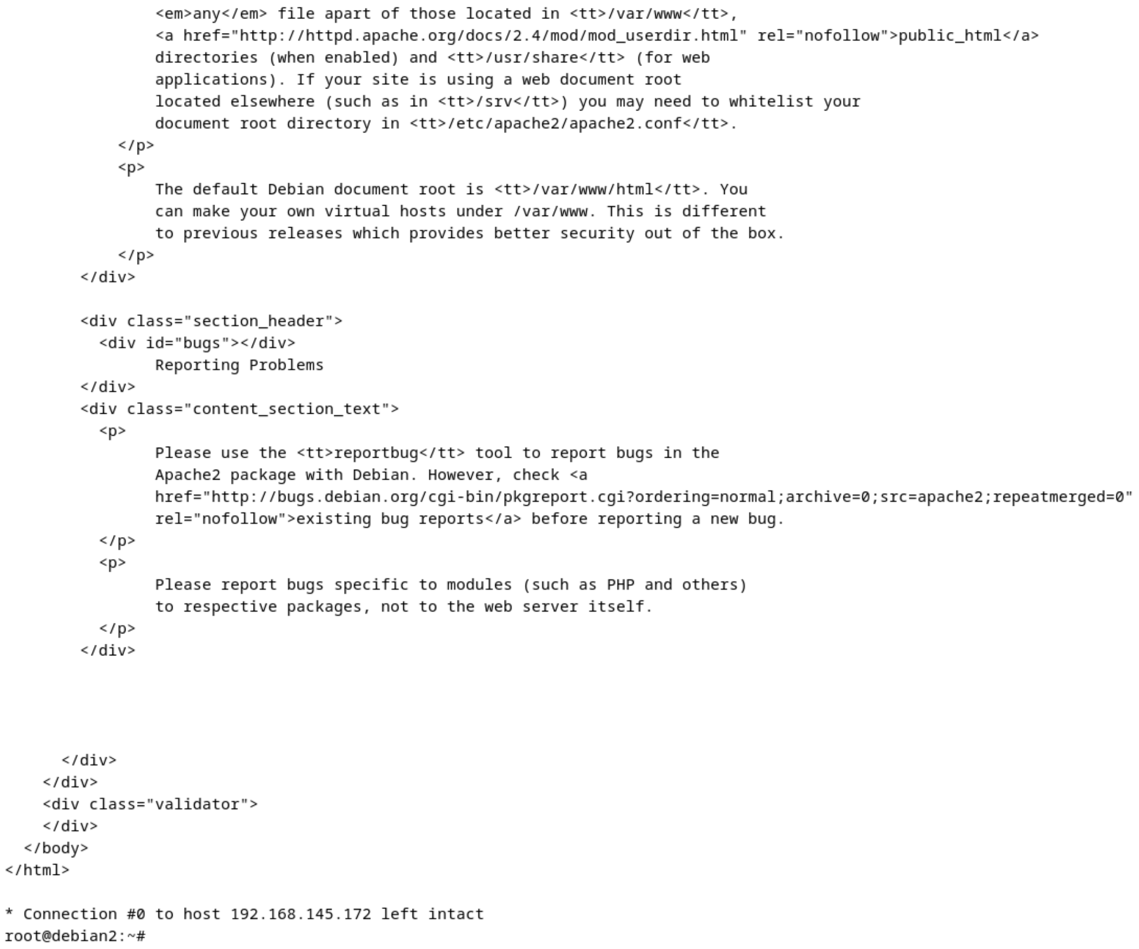
Le message indique bien :

1 added, 0 removed; done.

Cela signifie que **le certificat de la CA est maintenant bien pris en compte par le système** sur debian2.

Dernier étape : tester la connexion avec HTTPS2

curl -v https://192.168.145.172



```
                <em>any</em> file apart of those located in <tt>/var/www</tt>,
                <a href="http://httpd.apache.org/docs/2.4/mod/mod_userdir.html" rel="nofollow">public_html</a>
                directories (when enabled) and <tt>/usr/share</tt> (for web
                applications). If your site is using a web document root
                located elsewhere (such as in <tt>/srv</tt>) you may need to whitelist your
                document root directory in <tt>/etc/apache2/apache2.conf</tt>.
            </p>
            <p>
                The default Debian document root is <tt>/var/www/html</tt>. You
                can make your own virtual hosts under /var/www. This is different
                to previous releases which provides better security out of the box.
            </p>
        </div>

        <div class="section_header">
          <div id="bugs"></div>
                Reporting Problems
        </div>
        <div class="content_section_text">
          <p>
                Please use the <tt>reportbug</tt> tool to report bugs in the
                Apache2 package with Debian. However, check <a
                href="http://bugs.debian.org/cgi-bin/pkgreport.cgi?ordering=normal;archive=0;src=apache2;repeatmerged=0"
                rel="nofollow">existing bug reports</a> before reporting a new bug.
          </p>
          <p>
                Please report bugs specific to modules (such as PHP and others)
                to respective packages, not to the web server itself.
          </p>
        </div>




      </div>
    </div>
    <div class="validator">
    </div>
  </body>
</html>

* Connection #0 to host 192.168.145.172 left intact
root@debian2:~#
```