
SYSTEM SETUP AND LAB CONFIGURATION

Tanafaat Abdessamad

SOMMAIRE

Chapitre 1: Setup	1
Chapitre 2: SSH	2
Chapitre 3: SSL	3
Chapitre 4: Cronjob	4
Chapitre 5: Permissions	5
Chapitre 6: Webserver	6

Chapitre 1: Setup

Vérifier que Vagrant est installé.

Vérifier que les plugins Vagrant nécessaires sont installés

```
[→ ~ vagrant -v  
Vagrant 2.4.1  
[→ ~ vagrant plugin list  
vagrant-vmware-desktop (3.0.4, global)
```

```
cd ~/dev/labs/vm
```

```
[→ ~ vagrant -v  
Vagrant 2.4.1  
[→ ~ vagrant plugin list  
vagrant-vmware-desktop (3.0.4, global)  
[→ ~ mkdir -p ~/dev/labs/vm  
[→ ~ cd ~/dev/labs/vm  
[→ vm ls  
Vagrantfile
```

Démarrer la VM avec Vagrant :

```
[→ vm vagrant up --provider vmware_desktop  
Bringing machine 'default' up with 'vmware_desktop' provider...  
==> default: Machine is already running.  
[→ vm vagrant status  
Current machine states:  
  
default           running (vmware_desktop)  
  
The VM is running. To stop this VM, you can run `vagrant halt` to  
shut it down, or you can run `vagrant suspend` to simply suspend  
the virtual machine. In either case, to restart it again, run  
`vagrant up`.  
[→ vm ]
```

Chapitre 2: SSH

Se connecter en SSH à la machine

- Vérifier la configuration SSH de Vagrant :

```
[+ vm vagrant ssh-config
Host default
HostName 127.0.0.1
User vagrant
Port 2222
UserKnownHostsFile /dev/null
StrictHostKeyChecking no
PasswordAuthentication no
IdentityFile /Users/mac/dev/labs/vm/.vagrant/machines/default/vmware_desktop/p
rivate_key
IdentitiesOnly yes
LogLevel FATAL
PubkeyAcceptedKeyTypes +ssh-rsa
HostKeyAlgorithms +ssh-rsa

[+ vm ]
```

Se connecter en SSH avec un mot de passe :

```
[known]
[+ vm vagrant ssh
[vagrant@127.0.0.1's password:
Linux debian-11 5.10.0-31-amd64 #1 SMP Debian 5.10.221-1 (2024-07-14) x86_64

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento

Use of this system is acceptance of the OS vendor EULA and License Agreements.

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: Wed Feb  5 09:10:17 2025 from 172.16.168.1
vagrant@debian-11:~$ ]
```

Vérifier que les clés existent et les permissions des clés SSH :

```
[+ vm ls ~/.ssh/vm_key ~/.ssh/vm_key.pub
/Users/mac/.ssh/vm_key      /Users/mac/.ssh/vm_key.pub
[+ vm ls -l ~/.ssh/vm_key ~/.ssh/vm_key.pub
-rw----- 1 mac  staff  1843 Feb  3 17:08 /Users/mac/.ssh/vm_key
-rw-r--r-- 1 mac  staff   410 Feb  3 17:08 /Users/mac/.ssh/vm_key.pub
[+ vm ]
```

Vérifier l'ajout de la clé publique dans la machine virtuelle :

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Feb  5 09:11:11 2025 from 172.16.168.1
[vagrant@debian-11:~$ ls -l ~/.ssh/authorized_keys
-rw----- 1 vagrant vagrant 410 Feb  4 08:53 /home/vagrant/.ssh/authorized_keys
[vagrant@debian-11:~$ ]
```

Vérifier les permissions du fichier authorized_keys :

```
[vagrant@debian-11:~$ ls -l ~/.ssh/authorized_keys
-rw----- 1 vagrant vagrant 410 Feb  4 08:53 /home/vagrant/.ssh/authorized_keys
[vagrant@debian-11:~$ cat ~/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDW8pN9E01ZLHDY+WV+CPOxEP6iO+Plpj6IIirvWTC
uiH2j/rVl05+ttsITVaU+nkuto1UhMZLBq9wQuD9+dqvGd+B307CcAfEAKwHUxdRrUoho7S5WnEHfKsS
2tRBQwYUPtgHUDn3A9h1LnBGHyJNYOWh0Ur/A+TrCZigKWyBYip6WnTwMy+eylhiNnKzkNKPOLW87IM
5MRJGN570j/KTB8f5s9U/hPTJTeCVkZp2Xn900RNZjnq1dWnjujUM9AmWUzdljAgYpHrVRA4Xp1MdmGc
i52UcQGQo7BuvGMJV0ayqZOYIkXy8kZxW7jsxYvP5NpMVSWayFyMY1T/AQYx mac@MacBook-Pro-de-
mac.local
[vagrant@debian-11:~$ ]
```

Vérifier les permissions du fichier authorized_keys :

```
[vagrant@debian-11:~$ ls -l ~/.ssh/authorized_keys
-rw----- 1 vagrant vagrant 410 Feb  4 08:53 /home/vagrant/.ssh/authorized_keys
vagrant@debian-11:~$ ]
```

l'accès SSH sans mot de passe :

```
[→ vm ssh -p 2222 vagrant@127.0.0.1
Linux debian-11 5.10.0-31-amd64 #1 SMP Debian 5.10.221-1 (2024-07-14) x86_64

This system is built by the Bento project by Chef Software
More information can be found at https://github.com/chef/bento

Use of this system is acceptance of the OS vendor EULA and License Agreements.

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: Wed Feb  5 09:26:05 2025 from 172.16.168.1
vagrant@debian-11:~$ ]
```

- Vagrant redirige la connexion SSH depuis 127.0.0.1 (localhost) vers la machine virtuelle réelle.
- La machine virtuelle elle-même dispose d'une adresse IP interne différente attribuée par VMware (dans ce cas, 172.16.168.1).
-

Que se passe-t-il lorsque vous vous connectez en SSH à 127.0.0.1 ?

- Vagrant configure automatiquement la redirection de port entre la machine locale (notre hôte) et la machine virtuelle (VM).
- Vagrant démarre la VM et configure une connexion SSH.
- Il crée un tunnel SSH local depuis 127.0.0.1 (votre machine) vers l'IP interne de la VM en utilisant un port différent (par exemple, 2222).
- La commande SSH (ssh vagrant@127.0.0.1) se connecte à ce port redirigé.
- Vagrant redirige cette connexion vers la VM en arrière-plan.
- Cela signifie :
- Vous vous connectez en SSH à 127.0.0.1, mais Vagrant redirige cette demande vers la VM via la redirection de port.
- La VM elle-même ne voit pas 127.0.0.1 ; elle voit plutôt l'adresse IP réelle de votre machine (172.16.168.1).
-

Alternative : se connecter directement à la VM ?

- Au lieu de passer par 127.0.0.1, nous pouvons nous connecter directement à l'IP interne de la VM (si la mise en réseau le permet) :
- ssh vagrant@172.16.168.128 (IP de la VM)
- Mais cela ne fonctionne que si votre VM autorise l'accès SSH direct.

```
[vagrant@debian-11:~$ who
vagrant  tty1          Feb  4 09:08
vagrant  pts/0          Feb  4 17:07 (172.16.168.1)
vagrant  pts/1          Feb  5 09:52 (172.16.168.1)
vagrant@debian-11:~$ ]
```

Vérifiez l'IP de la VM : 172.16.168.128

```
[vagrant@debian-11:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:90:9f:da brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname ens160
    inet 172.16.168.128/24 brd 172.16.168.255 scope global dynamic eth0
        valid_lft 1488sec preferred_lft 1488sec
    inet6 fe80::20c:29ff:fe90:9fda/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 00:0c:29:90:9f:e4 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
```

Chapitre 3: SSL

```
vm — mac@MacBook-Pro-de-mac — ~/dev/labs/vm — -zsh — 80x24
[→ vm curl -v https://expired.badssl.com/
* Host expired.badssl.com:443 was resolved.
* IPv6: (none)
* IPv4: 104.154.89.105
* Trying 104.154.89.105:443...
* Connected to expired.badssl.com (104.154.89.105) port 443
* ALPN: curl offers h2,http/1.1
* (304) (OUT), TLS handshake, Client hello (1):
* CAfile: /etc/ssl/cert.pem
* CAPath: none
* (304) (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (OUT), TLS alert, certificate expired (557):
* SSL certificate problem: certificate has expired
* Closing connection
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (OUT), TLS alert, certificate expired (557):
curl: (60) SSL certificate problem: certificate has expired
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

Pour expired.badssl.com : un message d'erreur SSL/TLS indiquant que le certificat a expiré.

```
vm — mac@MacBook-Pro-de-mac — ~/dev/labs/vm — -zsh — 80x24
[→ vm curl -v https://self-signed.badssl.com/
* Host self-signed.badssl.com:443 was resolved.
* IPv6: (none)
* IPv4: 104.154.89.105
* Trying 104.154.89.105:443...
* Connected to self-signed.badssl.com (104.154.89.105) port 443
* ALPN: curl offers h2,http/1.1
* (304) (OUT), TLS handshake, Client hello (1):
* CAfile: /etc/ssl/cert.pem
* CAPath: none
* (304) (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (OUT), TLS alert, unknown CA (560):
* SSL certificate problem: self signed certificate
* Closing connection
* TLSv1.2 (IN), TLS handshake, Certificate (11):
* TLSv1.2 (OUT), TLS alert, unknown CA (560):
curl: (60) SSL certificate problem: self signed certificate
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
```

Pour self-signed.badssl.com : un message d'erreur SSL/TLS indiquant que le certificat est auto-signé et non approuvé par l'autorité de certification (CA).

L'erreur de certificat auto-signé indique que le certificat a été signé par lui-même (et non par une autorité de certification reconnue).

Inspecter les certificats avec openssl :

Pour inspecter les certificats plus en détail et comprendre les problèmes, on utilise la commande suivante :

```
vm — mac@MacBook-Pro-de-mac — ~/dev/labs/vm — -zsh — 80x24
how to fix it, please visit the web page mentioned above.
[→ vm openssl s_client -connect expired.badssl.com:443
Connecting to 104.154.89.105
CONNECTED(00000005)
depth=2 C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Certification Authority
verify return:1
depth=1 C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Domain Validation Secure Server CA
verify return:1
depth=0 OU=Domain Control Validated, OU=PositiveSSL Wildcard, CN=*.badssl.com
verify error:num=10:certificate has expired
notAfter=Apr 12 23:59:59 2015 GMT
verify return:1
depth=0 OU=Domain Control Validated, OU=PositiveSSL Wildcard, CN=*.badssl.com
notAfter=Apr 12 23:59:59 2015 GMT
verify return:1
---
Certificate chain
0 s:OU=Domain Control Validated, OU=PositiveSSL Wildcard, CN=*.badssl.com
    i:C=GB, ST=Greater Manchester, L=Salford, O=COMODO CA Limited, CN=COMODO RSA Domain Validation Secure Server CA
        a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
        v:NotBefore: Apr  9 00:00:00 2015 GMT; NotAfter: Apr 12 23:59:59 2015 GMT
```

```
vm — mac@MacBook-Pro-de-mac — ~/dev/labs/vm — -zsh — 80x24
v:NotBefore: May 30 10:48:38 2000 GMT; NotAfter: May 30 10:48:38 2020 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIFSzCCBDOgAwIBAgIQSueVSfqavj8QDxekeOFpCTANBgkqhkiG9w0BAQsFADCB
kDELMAkGA1UEBhMCR0IxGzAZBgNVBqTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEaMBgGA1UEChMRQ09NT0RPIENBIExpbWl0ZWQxNjA0BgNV
BAMTLUNPTU9ETyBSU0EgRG9tYWluIFZhbgkYXRpb24gU2VjdXJlIFNlcnZlciBD
QTAeFw0xNTA0MDkwMDAwMDBaFw0xNTA0MTIyMzU5NTlaMFkxitAfBgNVBAsTGERV
bWFpbibDdb250cm9sIFZhbgkYXR1ZDEdMBsGA1UECxMUUG9zaXRpdmtVU0wgV2ls
ZGNhcqmQxFATBqNVBAMUDCouYmfkc3NsLmNvbTCCASiwdDQYJKoZIhvCNQEBBQAD
ggEPADCCAQoCggEBAMIE7PiM7gTCs9hQ1XBYzJMY61yoaEmwIrX51Z6xKyx2PmzA
S2BMT0qytMAPgLw+XLJhgL5XEFdEyt/ccRLvOmUL1A3pmccYYz2QULFRtMWhyef
dOsKnRFSJiFzbIRMeVXk0WvoBj1IFVktsyjbqv9u/2CVSndrOfEk0TG23U3AxPxT
uW1CrbV8/q71FdIzS0ciccfCFHpsK0o3St/qbLVytH5aohbcabFXRNsKEqveww9H
dFxBIuGa+RuT5q0iBikusbpJHawnnp7i/dAcgCskgjZjFeeU4EFy+b+a1SYQCeF
xxC7c3DvaRhBB0VVfPlkPz0sw61865MaTiBryoUCAwEAAaOCAdUwggHRMB8GA1Ud
IwQYMBaAFJCvajqUWgvYk0oSVnPkQ7Q6KNrnMB0GA1UdDgQWBBSd7sF7gQs6R21x
GH0RN508pRs/+zAOBgNVHQ8BAf8EBAMCBaAwDAYDVR0TAQH/BAIwADAdBgNVHSUE
FjAUBggrBgEFBQcDAQYIKwYBBQUHawIwTwYDVR0gBEgwRjA6BgsrBgeEAbiXAQIC
BzArMCKGCCsGAQUFBwIBFh1odHRwczoV3N1Y3VyzS5jb21vZG8uY29tL0NQUzAI
BgZngQwBAGewVAYDVR0fBE0wSzBjoEegRYZDaHR0cDovL2NyBC5jb21vZG9jYS5j
b20vQ09NT0RPULNBrg9tYWluVmFsaWRhdGlvbln1Y3VyzVNlcnZlckNBLmNybdCB
hQYIKwYBBQUHAQEeTB3ME8GCCsGAQUFBzACHkNodHRwOibvY3J0LmNvbW9kb2Nh
-----
```

```
SSL handshake has read 5003 bytes and written 455 bytes
Verification error: certificate has expired
---
```

```
verify return:1
depth=0 OU=Domain Control Validated, OU=PositiveSSL Wildcard, CN=*.badssl.com
verify error:num=10:certificate has expired
notAfter=Apr 12 23:59:59 2015 GMT
verify return:1
depth=0 OU=Domain Control Validated, OU=PositiveSSL Wildcard, CN=*.badssl.com
notAfter=Apr 12 23:59:59 2015 GMT
verify return:1
```

Corriger les problèmes ?

Certificat expiré :

- Solution : Renouvez le certificat expiré. Vous devrez obtenir un nouveau certificat auprès d'une autorité de certification (CA) et mettre à jour la configuration du serveur pour utiliser le nouveau certificat.
-

Certificat auto-signé :

- Solution : Utilisez un certificat d'une CA reconnue. La meilleure pratique consiste à obtenir un certificat d'une autorité de certification reconnue, comme Let's Encrypt ou une autre CA commerciale, et à l'installer sur le serveur.

Chapitre 4: CRON JOB

le répertoire et les fichiers temporaires :

```
→ ~ cd ~/temp
→ temp ls
file1.txt file2.txt file4.txt file6.txt file8.txt
file10.txt file3.txt file5.txt file7.txt file9.txt
→ temp
```

le content de script clean_temp.sh :

```
temp — nano ~/clean_temp.sh — nano — nano ~/clean_temp.sh — 80x24
UW PICO 5.09                               File: /Users/mac/clean_temp.sh

#!/bin/bash
LOG_FILE=~/cron_cleanup.log

echo "Running cleanup at $(date)" >> $LOG_FILE
find ~/temp -type f -mtime +7 -delete 2>> $LOG_FILE
```

Planifier l'exécution du script avec cron (tous les jours à minuit) :

Commande : **crontab -e**

```
temp — crontab -e — crontab — vi « crontab — 80x24
1 crontab.OYNvXrDwtE
0 0 * * * ~/clean_temp.sh
```

Commande pour rendre le script exécutable :

```
→ temp cd ..
→ ~ ls
Desktop           Public
Documents          Virtual Machines.localized
Downloads          clean_temp.sh
Library            cron_cleanup.log
Movies              dev
Music               maximum-awesome
Pictures           temp
→ ~ chmod +x ~/clean_temp.sh
```

Vérifier si le cronjob fonctionne ?

Pour faciliter la tâche, j'ai configuré le cron pour qu'il s'exécute toutes les 5 minutes afin de voir les logs.

```
→ ~ cat cron_cleanup.log
Running cleanup at Tue Feb  4 15:50:01 +01 2025
Running cleanup at Tue Feb  4 15:55:00 +01 2025
Running cleanup at Tue Feb  4 16:00:00 +01 2025
→ ~
```

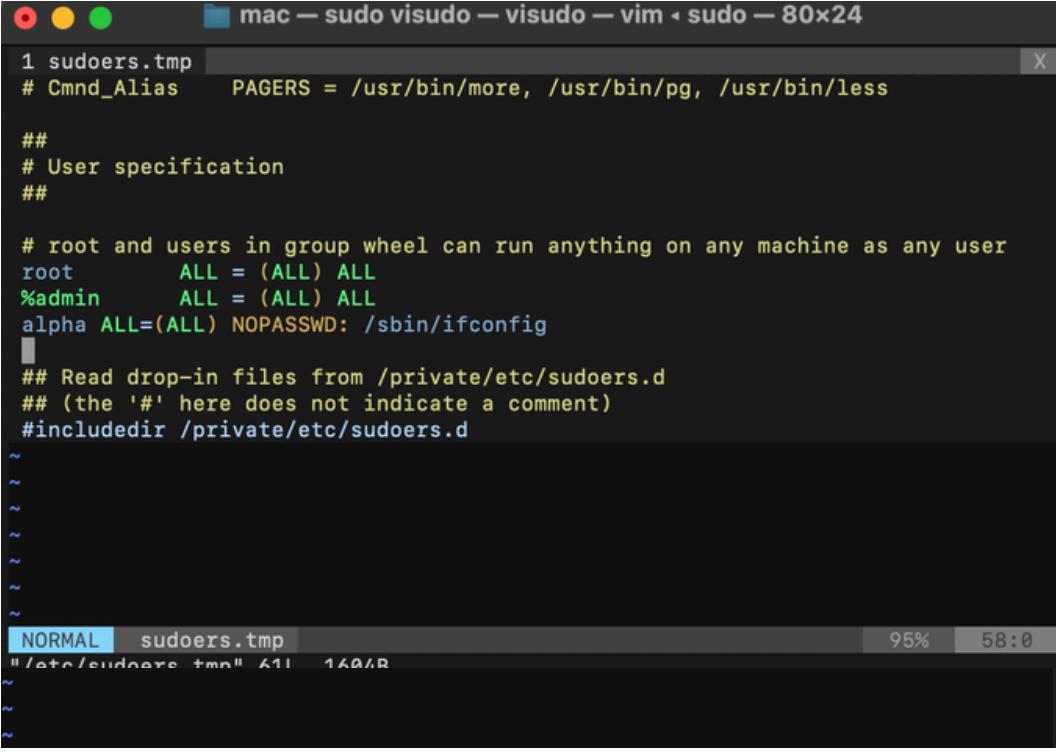
Chapitre 5 : Permissions

Modifier le fichier sudoers afin d'accorder à l'utilisateur alpha l'accès à la commande ifconfig sans lui donner d'autres privilèges administratifs.

Ajouter une règle pour accorder l'accès à **ifconfig** uniquement :

l'ajout de la ligne suivante à la fin du fichier **sudoers**:

alpha ALL=(ALL) NOPASSWD: /sbin/ifconfig



The screenshot shows a terminal window titled "mac — sudo visudo — visudo — vim • sudo — 80x24". The content of the file "sudoers.tmp" is displayed:

```
1 sudoers.tmp
# Cmnd_Alias      PAGERS = /usr/bin/more, /usr/bin/pg, /usr/bin/less

##
# User specification
##

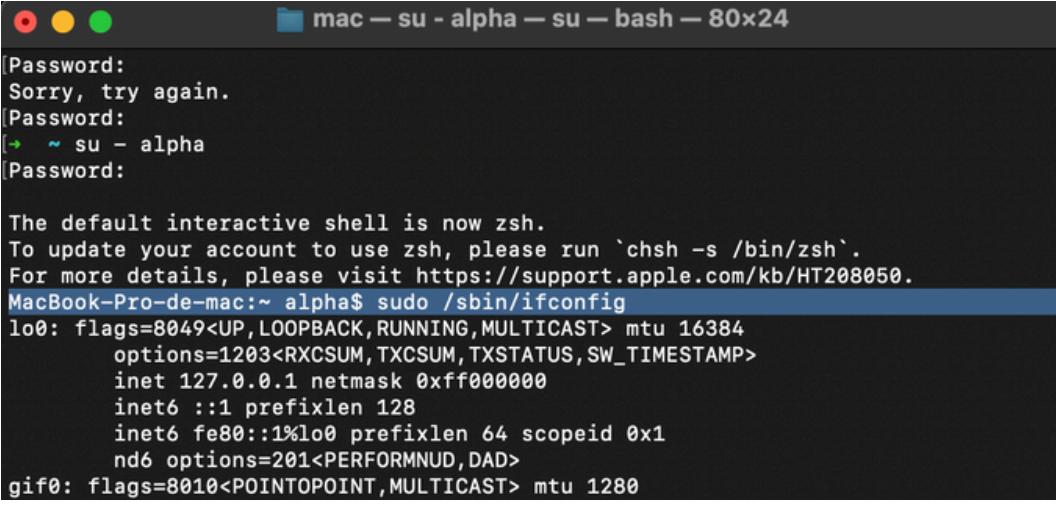
# root and users in group wheel can run anything on any machine as any user
root      ALL = (ALL) ALL
%admin    ALL = (ALL) ALL
alpha    ALL=(ALL) NOPASSWD: /sbin/ifconfig

##
## Read drop-in files from /private/etc/sudoers.d
## (the '#' here does not indicate a comment)
#include /private/etc/sudoers.d

~
~
~
~
~
~
NORMAL sudoers.tmp
"/etc/sudoers tmp" 611 1601R
```

Cette ligne donne à l'utilisateur alpha la possibilité d'exécuter la commande ifconfig en tant que superutilisateur (root) sans demander de mot de passe.

Vérifier la configuration avec su - alpha :



The screenshot shows a terminal window titled "mac — su - alpha — su — bash — 80x24". The session starts with password prompts for the alpha user:

```
[Password:
Sorry, try again.
[Password:
[+ ~ su - alpha
[Password:

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
```

Then, the user runs the sudo command followed by ifconfig:

```
MacBook-Pro-de-mac:~ alpha$ sudo /sbin/ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xffff0000
        inet6 ::1 prefixlen 128
            inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
                nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
```

La commande ifconfig s'exécute sans demander de mot de passe.

Problème : Pourquoi l'erreur "Permission denied" se produit-elle ?

```
MacBook-Pro-de-mac:~ alpha$ ls -ld /opt
drwxr-xr-x  6 root  wheel  192 Feb  4 17:24 /opt
[MacBook-Pro-de-mac:~ alpha$ curl -L -o /opt/bat.zip https://github.com/sharkdp/bat/archive/refs/tags/v0.24.0.zip
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent   Left  Speed
  0     0     0     0     0       0      0 --:--:-- --:--:-- --:--:--   0
Warning: Failed to open the file /opt/bat.zip: Permission denied
  0     0     0     0     0       0      0 --:--:--  0:00:01 --:--:--   0
curl: (23) Failure writing output to destination
MacBook-Pro-de-mac:~ alpha$
```

Cette erreur survient car l'utilisateur courant ALPHA n'a pas les permissions d'écriture nécessaires pour écrire dans le répertoire /opt.

Solution : Créer un sous-répertoire dans /opt/alpha et y donner les permissions nécessaires

```
[→ ~ sudo chown alpha: /opt/alpha
[→ ~ ls -ld /opt/alpha
drwxr-xr-x  4 alpha  wheel  128 Feb  4 17:47 /opt/alpha
[→ ~
```

télécharger le fichier bat.zip dans ce nouveau répertoire sans avoir de problèmes de permission :

```
[MacBook-Pro-de-mac:~ alpha$ curl -L -o /opt/alpha/bat.zip https://github.com/sharkdp/bat/archive/refs/tags/v0.24.0.zip
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload   Total Spent   Left  Speed
  0     0     0     0     0       0      0 --:--:-- --:--:-- --:--:--   0
100 2971k  0 2971k  0     0  1900k      0 --:--:--  0:00:01 --:--:-- 2496k
MacBook-Pro-de-mac:~ alpha$
```

décompresser un fichier .tar en utilisant la commande ditto :

```
MacBook-Pro-de-mac:alpha alpha$ ditto -xk /opt/alpha/bat.zip /opt/alpha/
MacBook-Pro-de-mac:alpha alpha$ cd /opt/alpha/
MacBook-Pro-de-mac:alpha alpha$ cd /opt/alpha
MacBook-Pro-de-mac:alpha alpha$ cd /opt/alpha/
MacBook-Pro-de-mac:alpha alpha$ ls
bat-0.24.0      bat.zip
MacBook-Pro-de-mac:alpha alpha$
```

Chapitre 6 : WebServer

Installation d'Apache et Vérification de l'état du service Apache2

```
● vm — vagrant@debian-11: ~ — ssh - vagrant ssh — 80x24
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.62-1~deb11u2).
0 upgraded, 0 newly installed, 0 to remove and 52 not upgraded.
[vagrant@debian-11:~$ sudo systemctl start apache2
[vagrant@debian-11:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Tue 2025-02-04 08:05:11 UTC; 1 day 3h ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 584 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUC>
   Process: 2426 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=>
 Main PID: 607 (apache2)
   Tasks: 55 (limit: 1110)
  Memory: 19.6M
    CPU: 5.349s
   CGroup: /system.slice/apache2.service
           └─ 607 /usr/sbin/apache2 -k start
             ├─2430 /usr/sbin/apache2 -k start
             └─2431 /usr/sbin/apache2 -k start

Warning: some journal files were not opened due to insufficient permissions.
lines 1-16/16 (END)
```

Active: active (running) indique que le service Apache2 fonctionne correctement.

Vérifier si Apache2 écoute sur le port 80

```
[vagrant@debian-11:~$ sudo ss -tulpn | grep :80
tcp  LISTEN  0      511          *:80          *:*      users:(("apache2
",pid=2431,fd=4),("apache2",pid=2430,fd=4),("apache2",pid=607,fd=4))
vagrant@debian-11:~$
```

Vérifier Apache2 avec un navigateur :

```
[vagrant@debian-11:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qlen 1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default
qlen 1000
  link/ether 00:0c:29:90:9f:da brd ff:ff:ff:ff:ff:ff
  altname enp3s0
  altname ens160
  inet 172.16.168.128/24 brd 172.16.168.255 scope global dynamic eth0
    valid_lft 1101sec preferred_lft 1101sec
```

