

Script Rsync

SOMMAIRE

Rsync c'est quoi ?

Installation du paquet Rsync

Première synchronisation en local

Synchronisation d'un serveur avec SSH

Génération d'une clé SSH

Copie de votre clé publique sur le serveur

Droits sur le répertoire .ssh/ et ses fichiers

Désactivation de la méthode d'authentification par mot de passe

Création du script de sauvegarde

Test du script de sauvegarde

Automatisation de la sauvegarde avec crontab

Connaître la taille du dossier de sauvegarde



RSYNC C'EST QUOI ?

RSYNC C'EST QUOI ?

Rsync est un outil qui permet de sauvegarder par copies vos données.

Il copie tous les fichiers sources vers la destination que vous avez choisie.

Ensuite, il ne mettra à jour que ceux qui ont été modifiés ou ajoutés.
D'où un gain de temps.

De plus, rsync peut aussi être utilisé à travers un réseau.

Sources:

<https://debian-facile.org/doc:reseau:rsync>

<https://technique.arscenic.org/transfert-de-donnees-entre/article/rsync-synchronisation-distant-de>

<https://technique.arscenic.org/connexion-distante-au-serveur-ssh/article/securisation-ssh-poussee-authentification-par-cle-rsa>

<https://www.linuxtricks.fr/wiki/cron-et-crontab-le-planificateur-de-taches>

INSTALLATION DU PAQUET RSYNC

INSTALLATION DU PAQUET RSYNC

Installation du paquet rsync sur le serveur de sauvegarde.

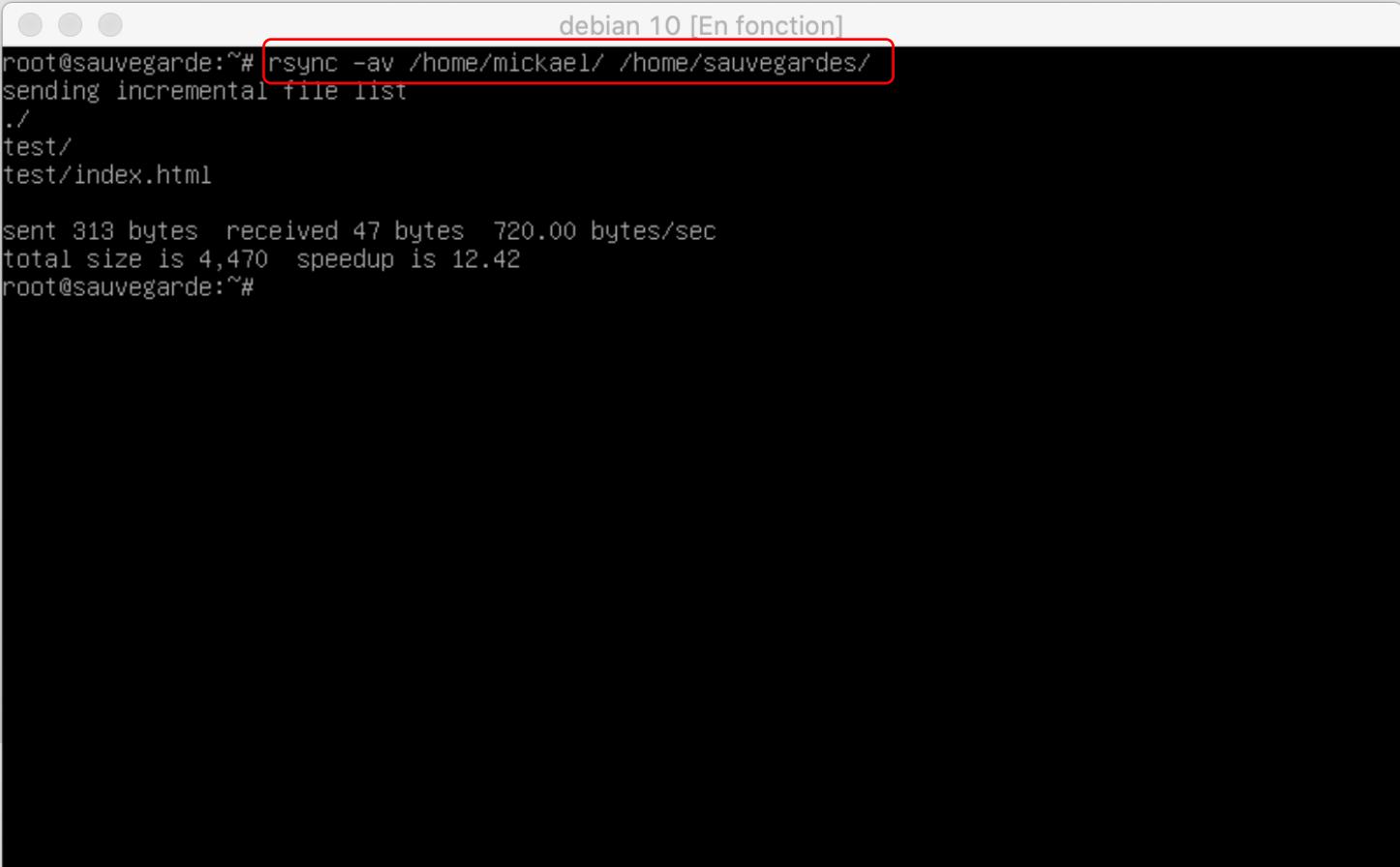
```
apt-get install rsync
```

PREMIÈRE SYNCHRONISATION EN LOCAL

PREMIÈRE SYNCHRONISATION EN LOCAL

Pour tester le principe du rsync on test dans un premier temps une copie sur son propre serveur (copie du dossier /home/mickael vers /home/sauvegardes).

```
rsync -r -avz /source/ /destination
```



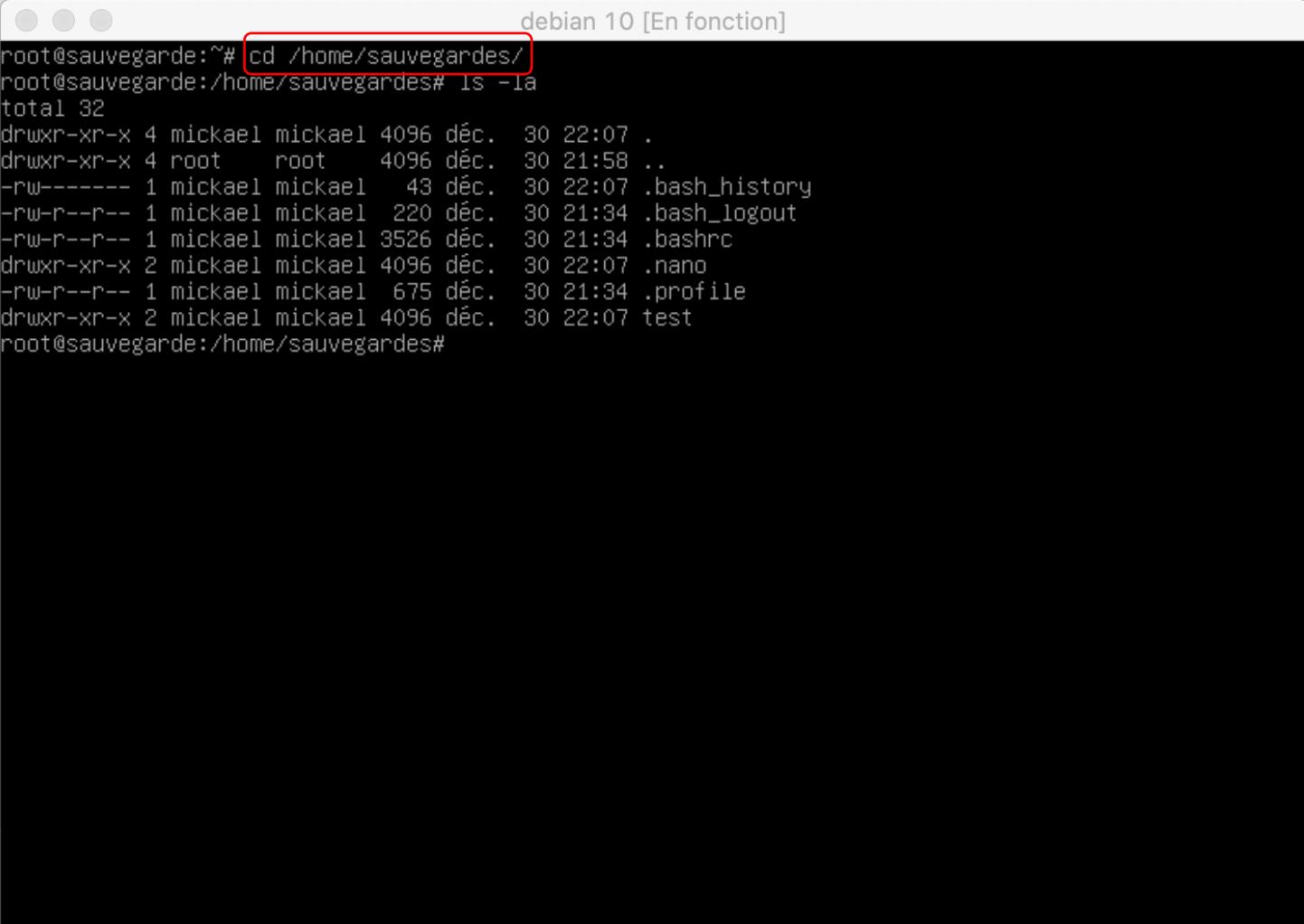
The screenshot shows a terminal window titled "debian 10 [En fonction]". The command entered is "rsync -av /home/mickael/ /home/sauvegardes/". The output shows the command being run, followed by a list of files transferred: "./", "test/", and "test/index.html". Performance metrics are displayed at the end: "sent 313 bytes received 47 bytes 720.00 bytes/sec", "total size is 4,470 speedup is 12.42", and the prompt "root@sauvegarde:~#". A red box highlights the command line.

```
root@sauvegarde:~# rsync -av /home/mickael/ /home/sauvegardes/
sending incremental file list
./
test/
test/index.html

sent 313 bytes received 47 bytes 720.00 bytes/sec
total size is 4,470 speedup is 12.42
root@sauvegarde:~#
```

PREMIÈRE SYNCHRONISATION EN LOCAL

On vérifie que la copie s'est correctement effectuée dans le dossier de destination.



The screenshot shows a terminal window with the title "debian 10 [En fonction]". The command "cd /home/sauvegardes/" is highlighted with a red rectangle. The output of the "ls -la" command is displayed, showing a directory structure with files like .bash_history, .bash_logout, .bashrc, .nano, and .profile, all owned by mickael/mickael with various timestamps.

```
root@sauvegarde:~# cd /home/sauvegardes/
root@sauvegarde:/home/sauvegardes# ls -la
total 32
drwxr-xr-x 4 mickael mickael 4096 déc. 30 22:07 .
drwxr-xr-x 4 root    root    4096 déc. 30 21:58 ..
-rw----- 1 mickael mickael  43 déc. 30 22:07 .bash_history
-rw-r--r-- 1 mickael mickael 220 déc. 30 21:34 .bash_logout
-rw-r--r-- 1 mickael mickael 3526 déc. 30 21:34 .bashrc
drwxr-xr-x 2 mickael mickael 4096 déc. 30 22:07 .nano
-rw-r--r-- 1 mickael mickael  675 déc. 30 21:34 .profile
drwxr-xr-x 2 mickael mickael 4096 déc. 30 22:07 test
root@sauvegarde:/home/sauvegardes#
```

SYNCHRONISATION D'UN SERVEUR AVEC SSH

SYNCHRONISATION D'UN SERVEUR AVEC SSH

On va utiliser rsync pour récupérer des sources d'un autre serveur.

Pour ce première essai il est préférable d'éviter toutes contraintes:

- Autoriser l'accès root,

- Laisser le port 22 par défaut.

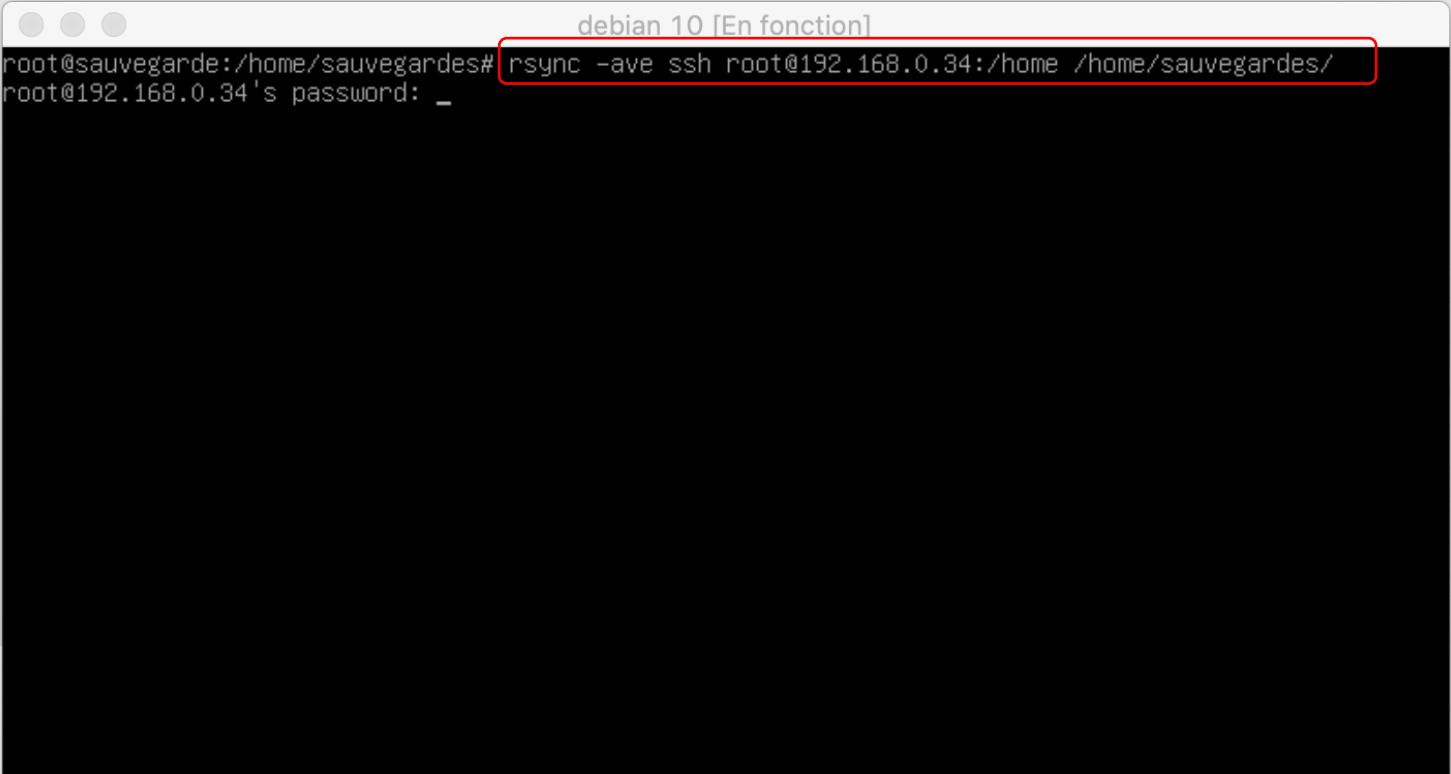
SYNCHRONISATION D'UN SERVEUR AVEC SSH

On tape la commande sur le serveur de sauvegarde.

```
rsync -ave ssh root@192.168.0.34:/home  
/home/destination
```

Si un port de connexion doit être spécifié

```
rsync -ave "ssh -p 1234" root@192.168.0.34:/home  
/home/destination
```



The screenshot shows a terminal window titled "debian 10 [En fonction]". The command "rsync -ave ssh root@192.168.0.34:/home /home/sauvegardes/" is entered at the prompt. The password prompt "root@192.168.0.34's password: " is visible below the command line. The entire command line is highlighted with a red rectangle.

```
debian 10 [En fonction]  
root@sauvegarde:/home/sauvegardes# rsync -ave ssh root@192.168.0.34:/home /home/sauvegardes/  
root@192.168.0.34's password: _
```

SYNCHRONISATION D'UN SERVEUR AVEC SSH

```
debian 10 [En fonction]
home/wordpress/www/wp-includes/sodium_compat/src/Core32/SecretStream/State.php
home/wordpress/www/wp-includes/sodium_compat/src/PHP52/
home/wordpress/www/wp-includes/sodium_compat/src/PHP52/SplFixedArray.php
home/wordpress/www/wp-includes/theme-compat/
home/wordpress/www/wp-includes/theme-compat/comments.php
home/wordpress/www/wp-includes/theme-compat/embed-404.php
home/wordpress/www/wp-includes/theme-compat/embed-content.php
home/wordpress/www/wp-includes/theme-compat/embed.php
home/wordpress/www/wp-includes/theme-compat/footer-embed.php
home/wordpress/www/wp-includes/theme-compat/footer.php
home/wordpress/www/wp-includes/theme-compat/header-embed.php
home/wordpress/www/wp-includes/theme-compat/header.php
home/wordpress/www/wp-includes/theme-compat/sidebar.php
home/wordpress/www/wp-includes/widgets/
home/wordpress/www/wp-includes/widgets/class-wp-nav-menu-widget.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-archives.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-calendar.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-categories.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-custom-html.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-links.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-media-audio.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-media-gallery.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-media-image.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-media-video.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-media.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-meta.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-pages.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-recent-comments.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-recent-posts.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-rss.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-search.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-tag-cloud.php
home/wordpress/www/wp-includes/widgets/class-wp-widget-text.php

sent 132,619 bytes received 196,233,243 bytes 15,709,268.96 bytes/sec
total size is 195,651,340 speedup is 1.00
root@sauvegarde:/home/sauvegardes#
```

GÉNÉRATION D'UNE CLÉ SSH

GÉNÉRATION D'UNE CLÉ SSH

Génération d'une clé avec le chiffrement RSA qui assure une sécurité à 99,8%.

https://fr.wikipedia.org/wiki/Chiffrement_RSA

L'idée est de créer une clé privée (crypté) qui sert à se connecter au serveur.

La clé doit être créée sur le serveur de sauvegarde.

Une clé publique va être générée dans le fichier suivant:

`~/ssh/id_rsa.pub`

GÉNÉRATION D'UNE CLÉ SSH

On tape la commande sur le serveur de sauvegarde.

```
ssh-keygen -t rsa
```

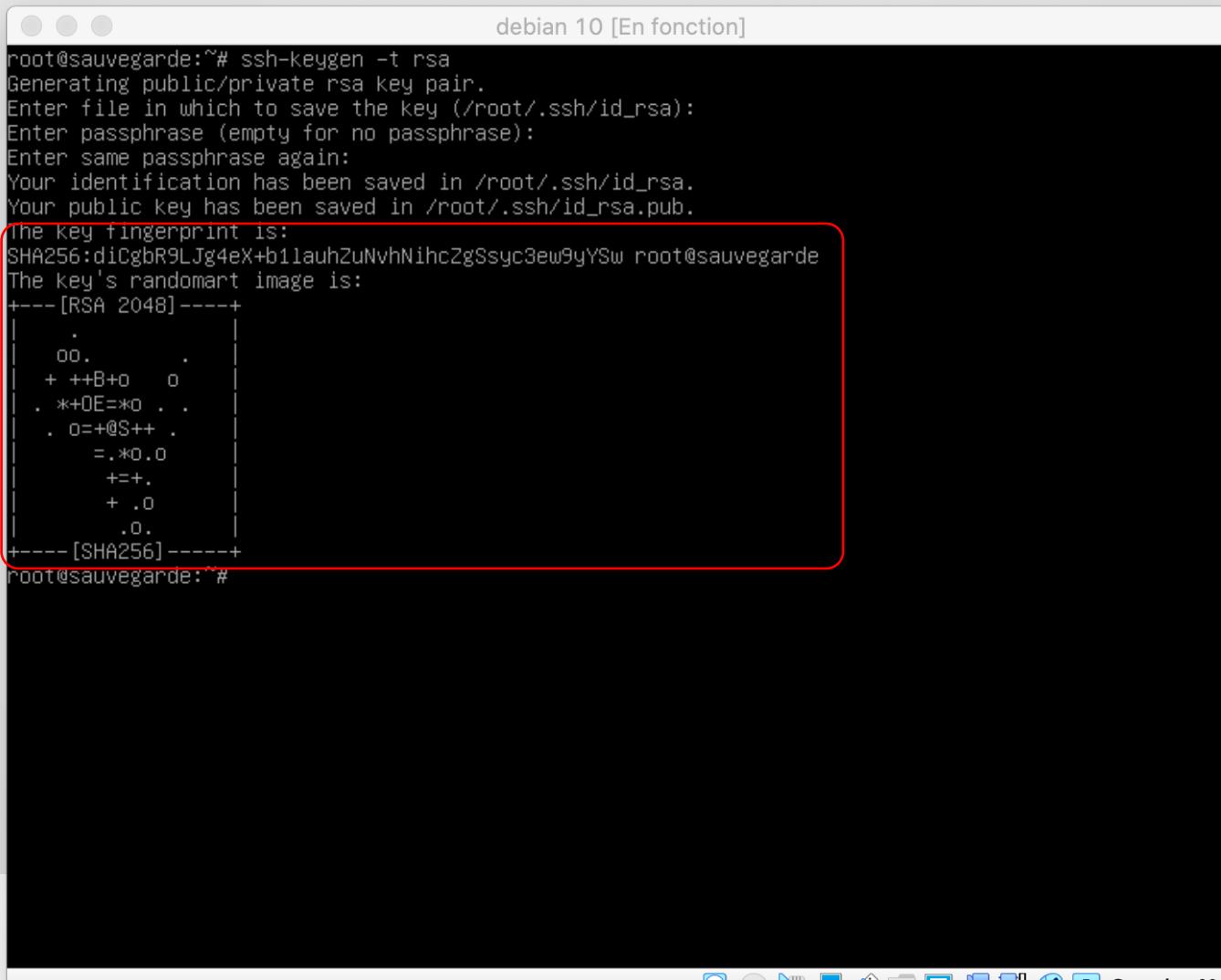
```
debain 10 [En fonction]
root@sauvegarde:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again: _
```

On fait
« entrer » pour
valider
l'enregistrement
de la clé dans
/root/.ssh/id_rsa

Dans notre cas,
n'entrez aucun
mot de passe. Si
vous entrez un
mot de passe il
devra être entré à
chaque demande
de connexion,
nous perdons le
bénéfice de
l'automatisation.

GÉNÉRATION D'UNE CLÉ SSH

La clé est générée dans /root/.ssh/id_rsa.pub



The screenshot shows a terminal window titled "debian 10 [En fonction]" with a black background and white text. The window contains the following output from the "ssh-keygen -t rsa" command:

```
root@sauvegarde:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:diCgbR9LJg4eX+b1lauh2UNvhNihcZgSsyc3ew9yYSw root@sauvegarde
The key's randomart image is:
+---[RSA 2048]---+
. oo.
+ ++B+o o
. **OE==o .
. o=+@S++ .
=.*o.o
+=+.
+ .o
. o.
+---[SHA256]---+
root@sauvegarde: ~#
```

A red rectangular box highlights the "randomart image" section of the output, which consists of a grid of characters representing the key's visual representation.

MODIFIER LE MOT DE PASSE

Il est possible de modifier le mot de passe:

```
ssh-keygen -p -t rsa
```



COPIE DE VOTRE CLÉ PUBLIQUE SUR
LE SERVEUR

COPIE DE VOTRE CLÉ PUBLIQUE SUR LE SERVEUR

Copie de votre clé publique sur le serveur (à sauvegarder).

La solution la plus simple consiste à utiliser la commande `ssh-copy-id` pour copier la clé sur le serveur.

Il est aussi possible d'effectuer cette copie de manière manuelle mais bien plus fastidieuse.

Il faut copier la clé sur le serveur de sauvegarde

`~/.ssh/id_rsa.pub`

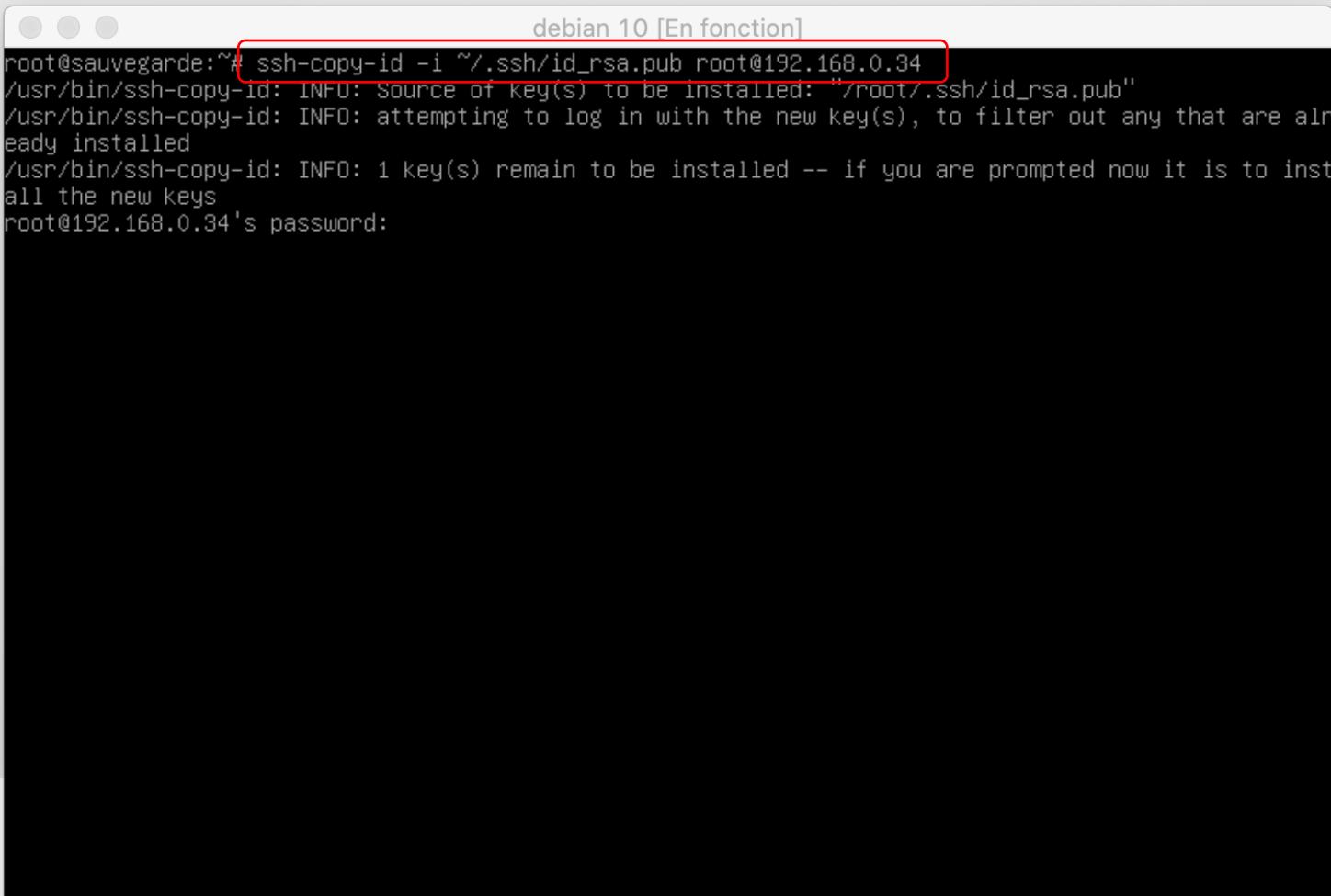
Puis la coller dans le serveur à sauvegarder

`~/.ssh/authorized_keys`

COPIE DE VOTRE CLÉ PUBLIQUE SUR LE SERVEUR

Copie de la clé SSH avec la commande

```
ssh-copy-id -i ~/.ssh/id_rsa.pub  
root@ip_serveur_a_sauvegarder
```



A terminal window titled "debian 10 [En fonction]" showing the execution of the `ssh-copy-id` command. The command is highlighted with a red box. The output shows the key being installed and a password prompt for the root user at the target IP address.

```
root@sauvegarde:~# ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.0.34  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
root@192.168.0.34's password:
```

Tapez le mot de passe root pour vous connecter au serveur.

COPIE DE VOTRE CLÉ PUBLIQUE SUR LE SERVEUR

```
debian 10 [En fonction]
root@sauvegarde:~# ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.0.34
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
root@192.168.0.34's password:
Number of key(s) added: 1

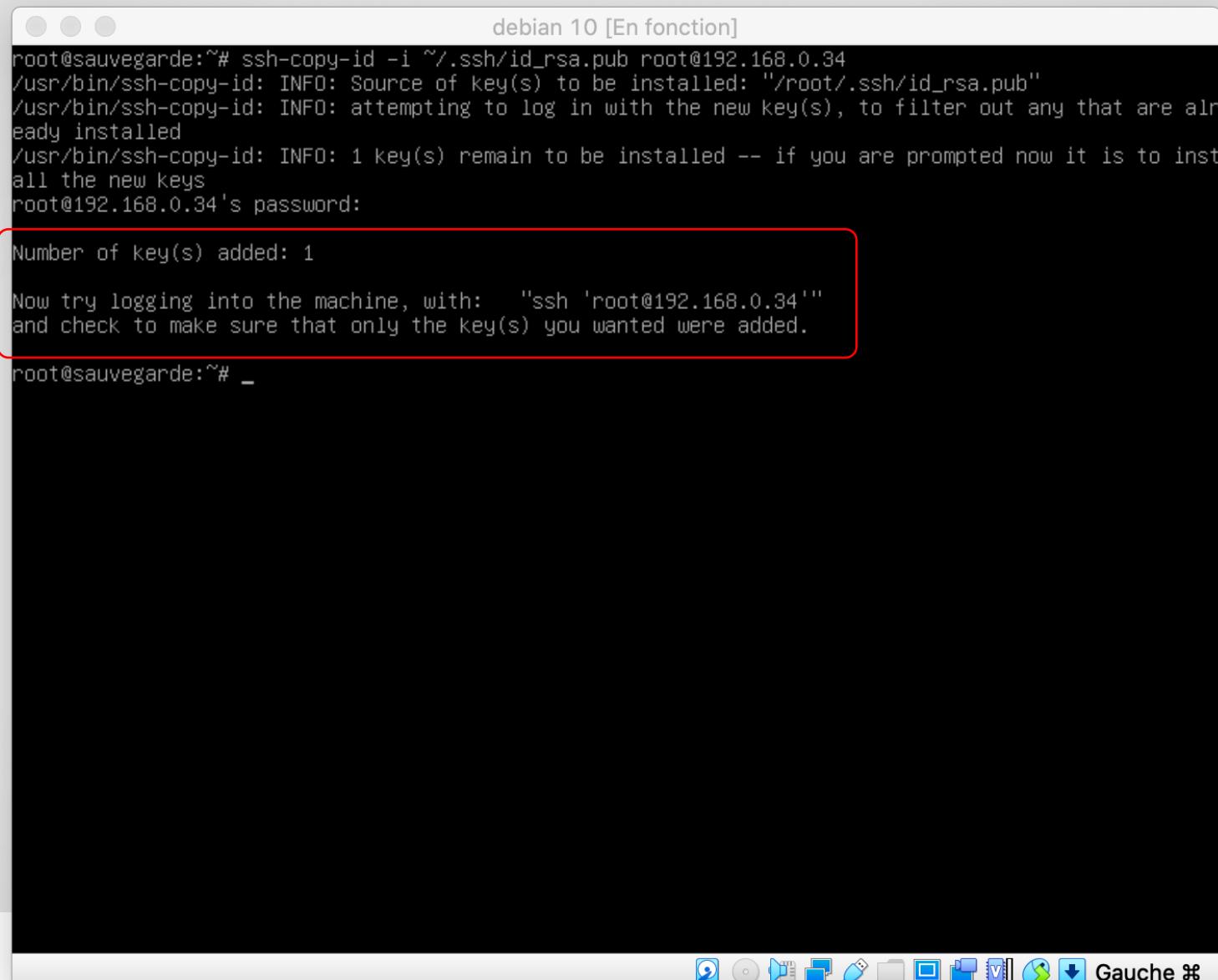
Now try logging into the machine, with: "ssh 'root@192.168.0.34'"
and check to make sure that only the key(s) you wanted were added.

root@sauvegarde:~# _
```

La clé est ajoutée.

Connectez-vous au serveur pour vérifier sur le serveur que la clé est bien ajoutée.

Dans:
nano
~/.ssh/authorized_keys



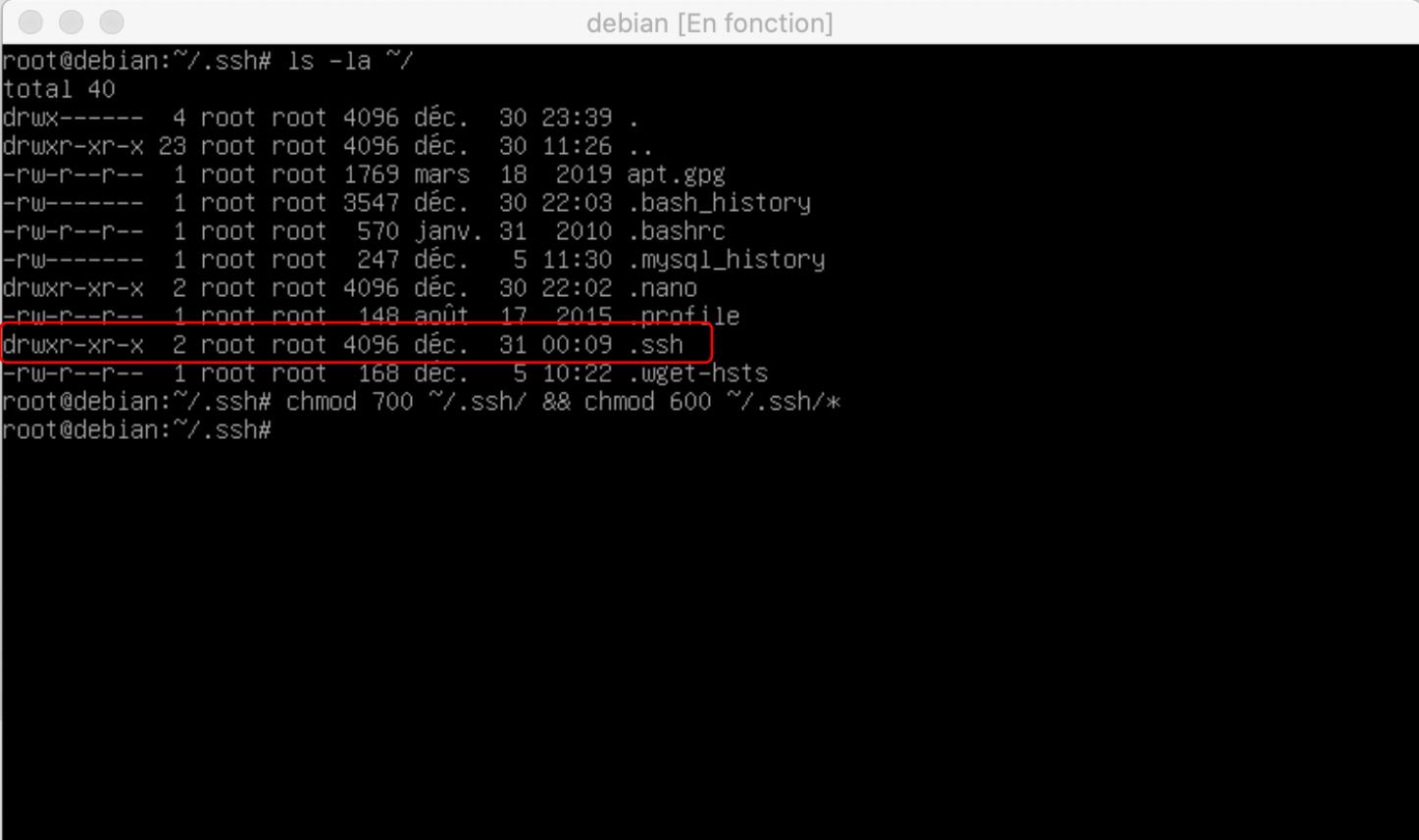
DROITS SUR LE RÉPERTOIRE .SSH/ ET SES FICHIERS

DROITS SUR LE RÉPERTOIRE .SSH/ ET SES FICHIERS

Il est important de sécuriser l'accès (du serveur à sauvegarder) au répertoire `~/.ssh/` et des fichiers qu'il contient.

Tapez la commande suivante:

```
chmod 700 ~/.ssh && chmod 600 ~/.ssh/*
```



```
debian [En fonction]
root@debian:~/.ssh# ls -la ~/
total 40
drwx----- 4 root root 4096 déc. 30 23:39 .
drwxr-xr-x 23 root root 4096 déc. 30 11:26 ..
-rw-r--r-- 1 root root 1769 mars 18 2019 apt.gpg
-rw----- 1 root root 3547 déc. 30 22:03 .bash_history
-rw-r--r-- 1 root root 570 janv. 31 2010 .bashrc
-rw----- 1 root root 247 déc. 5 11:30 .mysql_history
drwxr-xr-x 2 root root 4096 déc. 30 22:02 .nano
-rw-r--r-- 1 root root 148 août 17 2015 .profile
drwxr-xr-x 2 root root 4096 déc. 31 00:09 .ssh
-rw-r--r-- 1 root root 168 déc. 5 10:22 .wget-hsts
root@debian:~/.ssh# chmod 700 ~/.ssh/ && chmod 600 ~/.ssh/*
root@debian:~/.ssh#
```

DÉSACTIVATION DE LA MÉTHODE D'AUTHENTIFICATION PAR MOT DE PASSE

DÉSACTIVATION DE LA MÉTHODE D'AUTHENTIFICATION ...

On va désactiver la méthode d'authentification par mot de passe.

2 possibilités:

On peut désactiver la connexion SSH uniquement pour l'accès root

```
#PermitRootLogin prohibit-password
```

Ou désactiver la méthode d'authentification par mot de passe

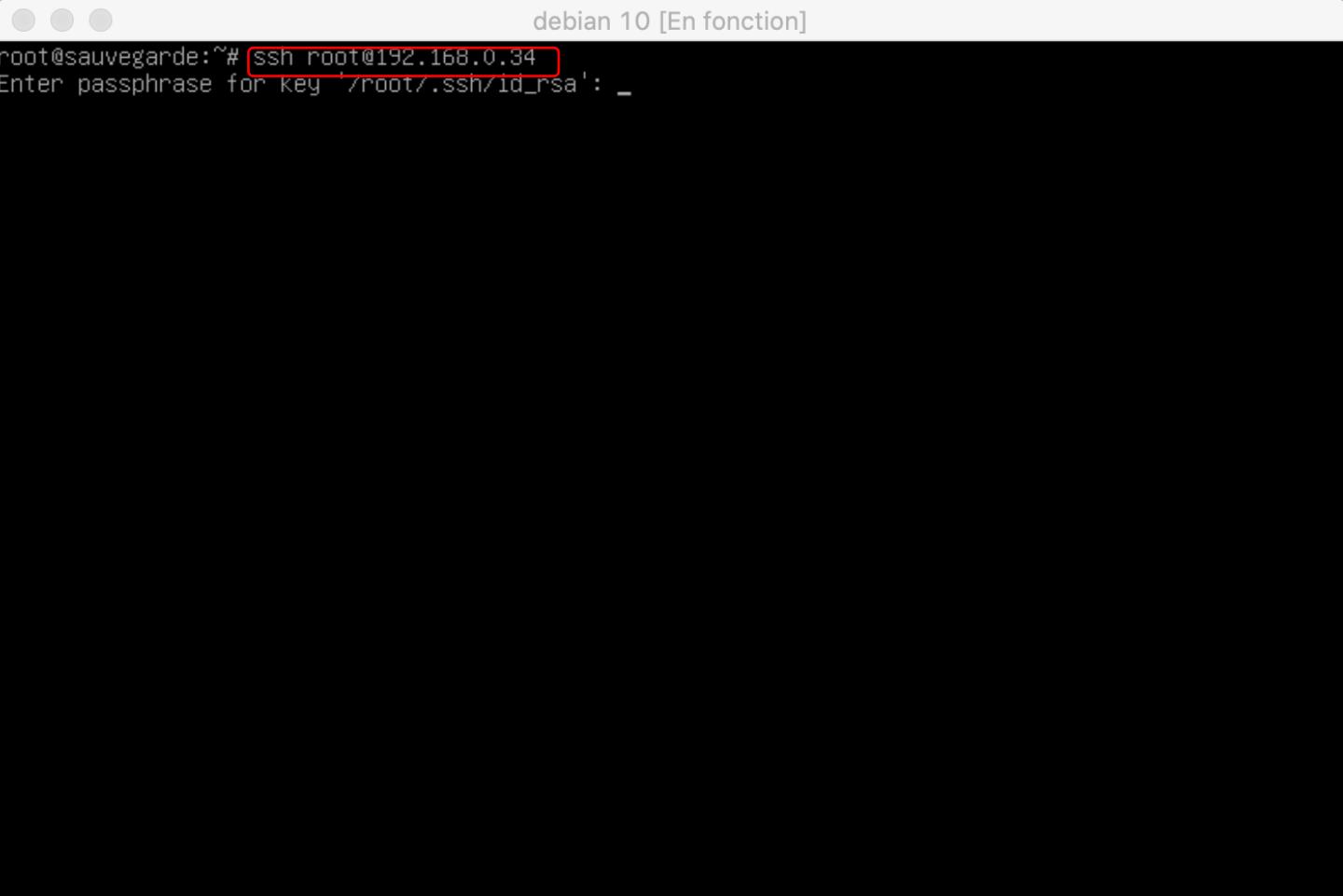
(totalement sécurisé car il n'est plus possible de se connecter sans avoir créé de clé crypté)

```
PasswordAuthentication no
```

Dorénavant seuls les utilisateurs autorisés à se connecter par clé pourront avoir accès au serveur et donc toute tentative d'intrusion étrangère sera rejetée.

DÉSACTIVATION DE LA MÉTHODE D'AUTHENTIFICATION ...

Lorsqu'on se connecte depuis notre serveur de sauvegarde, l'accès est possible avec le mot de passe indiqué lors de la création de notre clé RSA.

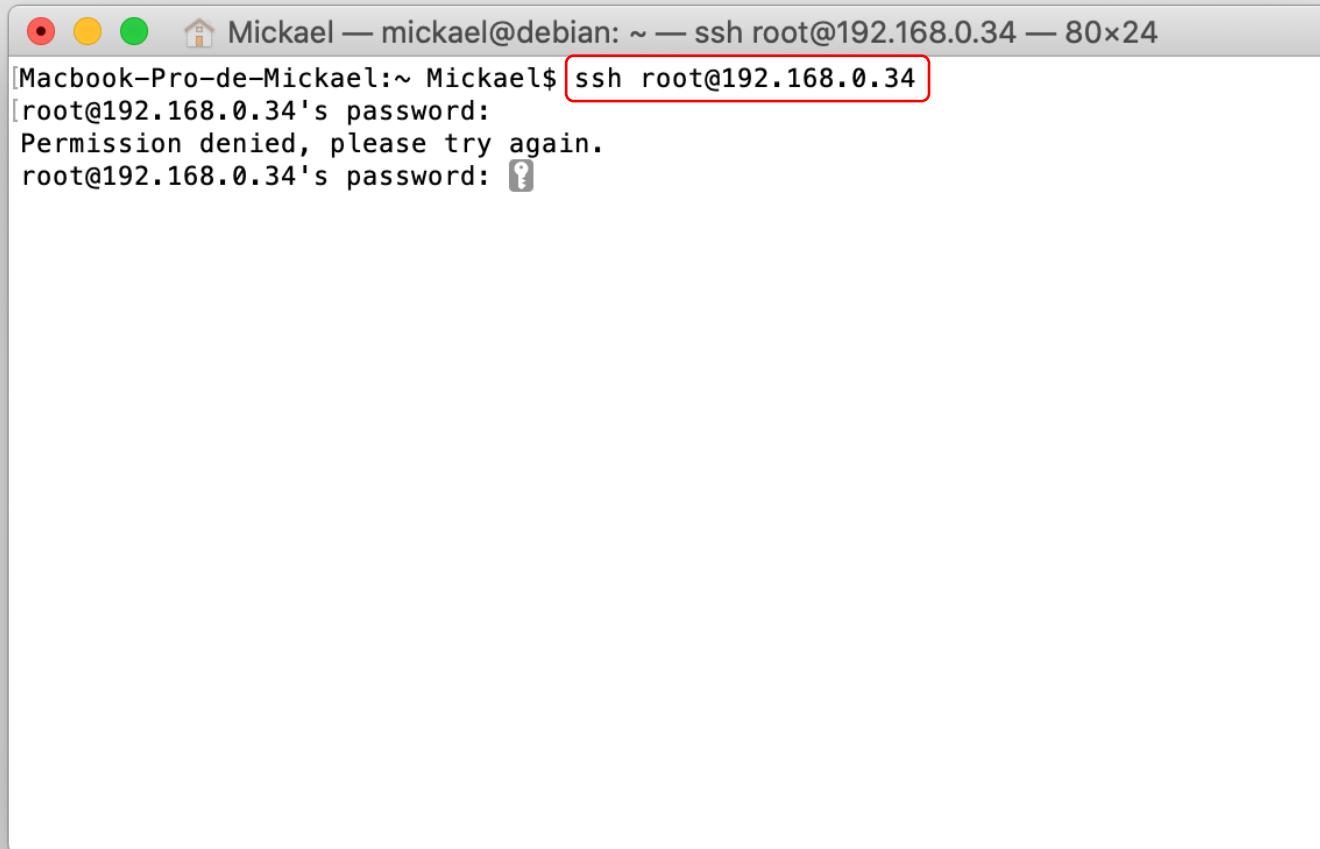


The screenshot shows a terminal window titled "debian 10 [En fonction]". The command entered is "ssh root@192.168.0.34", followed by a password prompt: "Enter passphrase for key '/root/.ssh/id_rsa':". The command and the password prompt are highlighted with a red rectangle.

```
root@sauvegarde:~# ssh root@192.168.0.34
Enter passphrase for key '/root/.ssh/id_rsa': _
```

DÉSACTIVATION DE LA MÉTHODE D'AUTHENTIFICATION ...

Si on se connecte via notre propre ordinateur l'accès est impossible.



```
Mickael — mickael@debian: ~ — ssh root@192.168.0.34 — 80x24
Macbook-Pro-de-Mickael:~ Mickael$ ssh root@192.168.0.34
root@192.168.0.34's password:
Permission denied, please try again.
root@192.168.0.34's password:
```

SAUVEGARDE MANUELLE À DISTANCE

SAUVEGARDE MANUELLE À DISTANCE

Utilisation de la commande Rsync manuellement en précisant un numéro de port.

```
rsync -ave "ssh -p 1234" root@192.168.0.31:/home  
/home/destination
```

CRÉATION DU SCRIPT DE SAUVEGARDE

CRÉATION DU SCRIPT DE SAUVEGARDE

Création du fichier script (à créer sur le serveur de sauvegarde):

```
nano /usr/local/scripts/sauvegarde_server1.sh
```

```
#!/bin/bash
# Adresse ip du serveur à sauvegarder
SERVEUR="192.168.0.19"

# Informations de sauvegarde
DST="/home/sauvegardes"
DATE=`date +%y%m%d-%H%M`
for REPE in etc home root var
do
rsync -avr --filter "- *site/tmp*" --delete -e "ssh -p 1234" $SERVEUR:$REPE $DST/current
done
cp -al $DST/current $DST/$DATE
```

Chemin de destination des sauvegardes.

Dossiers à sauvegarder.

Crée une copie dans le dossier /current puis les suivantes par date.

Pour la connexion SSH, si nécessaire, indiquez un port de connexion différent.

CRÉATION DU SCRIPT DE SAUVEGARDE

Création du fichier script (à créer sur le serveur de sauvegarde):

```
nano /usr/local/scripts/sauvegarde_server1.sh
```

```
#!/bin/bash

# Adresse ip du serveur à sauvegarder
SERVEUR="213.161.203.33"

# Informations de sauvegarde
DST="/home/sauvegardes"
DATE=`date +%y%m%d-%H%M`
for REPE in etc home root var
do
rsync -avr --filter "- *site/tmp*" --delete -e "ssh -p 1234"
$SERVEUR:$REPE $DST/current
done
cp -al $DST/current $DST/$DATE
```

CRÉATION DU SCRIPT DE SAUVEGARDE

On peut ajouter l'historique de sauvegarde (pour conserver uniquement les 7 derniers jours).

```
# Historique de sauvegarde sur 7 Jours
for i in 7 8
do
    rm -Rf $DST/`date --date $i' days ago' "+%y%m%d-*"`
done
```

TEST DU SCRIPT DE SAUVEGARDE

TEST DU SCRIPT DE SAUVEGARDE

Sur le serveur de sauvegarde, lancez le script de sauvegarde.
`/usr/local/scripts/sauvegarde-server1.sh`

```
debian 10 [En fonction]
root@sauvegarde:~# /usr/local/scripts/sauvegarde_server1.sh
receiving incremental file list

sent 245 bytes received 77,579 bytes 155,648.00 bytes/sec
total size is 3,581,865 speedup is 46.03
receiving incremental file list

sent 1,390 bytes received 311,397 bytes 625,574.00 bytes/sec
total size is 195,651,414 speedup is 625.51
receiving incremental file list

sent 43 bytes received 333 bytes 752.00 bytes/sec
total size is 6,918 speedup is 18.40
receiving incremental file list
var/lib/systemd/clock
var/log/auth.log
var/log/daemon.log
var/log/syslog
var/log/exim4/mainlog

sent 1,279 bytes received 106,961 bytes 216,480.00 bytes/sec
total size is 423,070,553 speedup is 3,908.63
root@sauvegarde:~#
```

TEST DU SCRIPT DE SAUVEGARDE

Si le script de sauvegarde ne se lance pas.

Vérifier si les droits d'exécutions sont activés.

```
chmod 766 sauvegarder_server1.sh
```

AUTOMATISATION DE LA SAUVEGARDE AVEC CRONTAB

AUTOMATISATION DE LA SAUVEGARDE AVEC CRONTAB

Crontab est un outil qui permet de lancer des applications de façon régulière, pratique pour un serveur pour y lancer des scripts de sauvegardes, etc...

SYNTAXE CRONTAB

Syntaxe de crontab (dans l'ordre de saisis dans le fichier crontab).

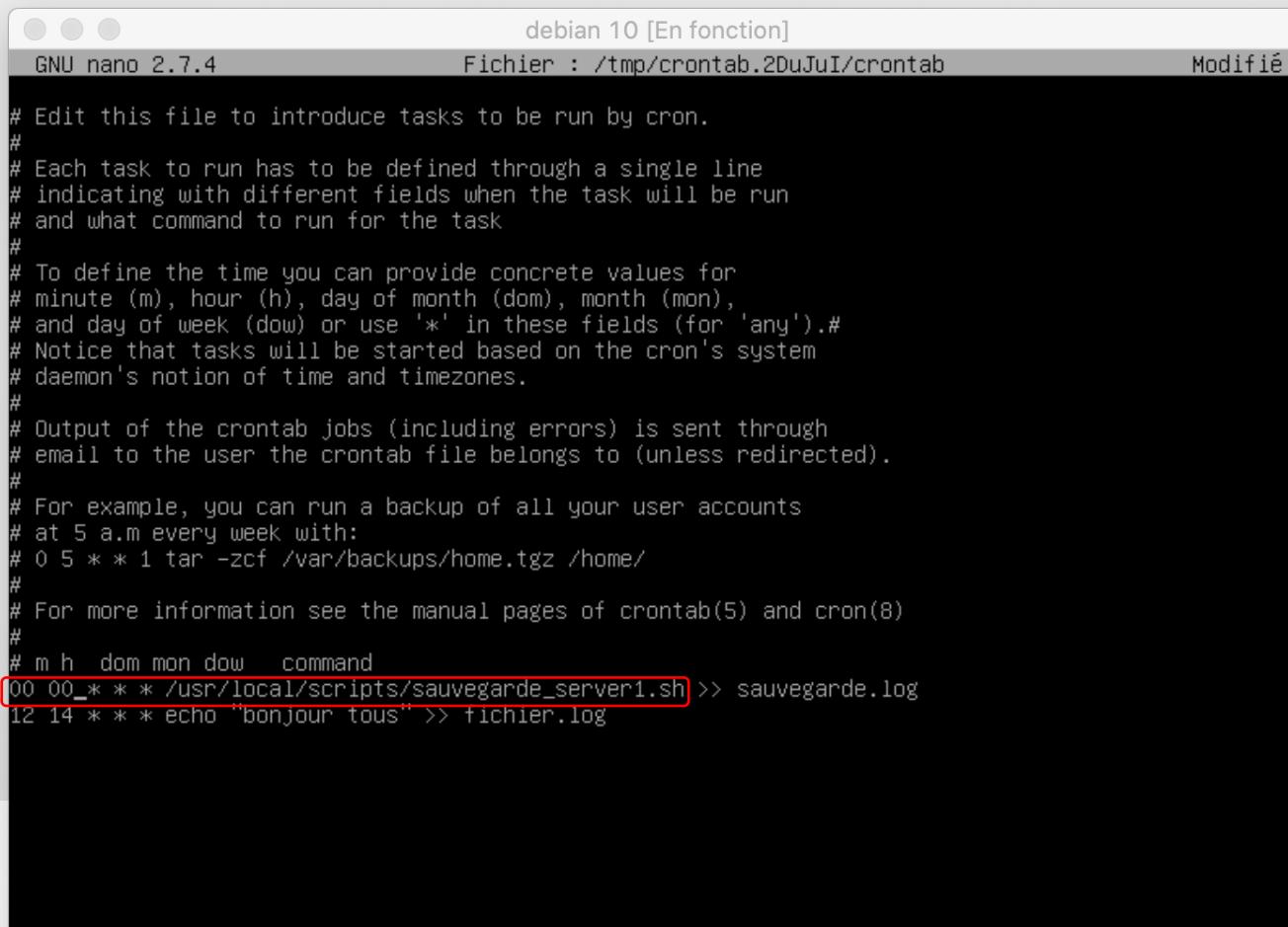
Minute	Heure	Jour du mois	Mois	Jour de la semaine	Commande
0-59	0-23	1-31	1-12	0-6	Bash script

AUTOMATISATION DE LA SAUVEGARDE AVEC CRONTAB

Pour configurer Crontab, éditez le fichier en tapant:

```
crontab -e
```

Dans notre exemple le script est exécuté tous les jours à 00h00.



The screenshot shows a terminal window titled "debian 10 [En fonction]" with the command "GNU nano 2.7.4". The file path is "Fichier : /tmp/crontab.2DuJuI/crontab" and the status is "Modifié". The content of the crontab file is as follows:

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
00 00_* * * */usr/local/scripts/sauvegarde_server1.sh >> sauvegarde.log  
12 14 * * * echo "bonjour tous" >> fichier.log
```

An annotation on the left side of the terminal window points to the second line of the crontab content with the text: "La 2^{ème} ligne permet de faire un test. Le résultat s'affiche dans le fichier.log". An arrow points from this text to the line "00 00_* * * */usr/local/scripts/sauvegarde_server1.sh >> sauvegarde.log". Another annotation on the right side of the terminal window points to the last line of the crontab content with the text: "Créer un fichier .log est toujours utilisé pour savoir si la commande s'est correctement exécuté". An arrow points from this text to the line "12 14 * * * echo \"bonjour tous\" >> fichier.log".

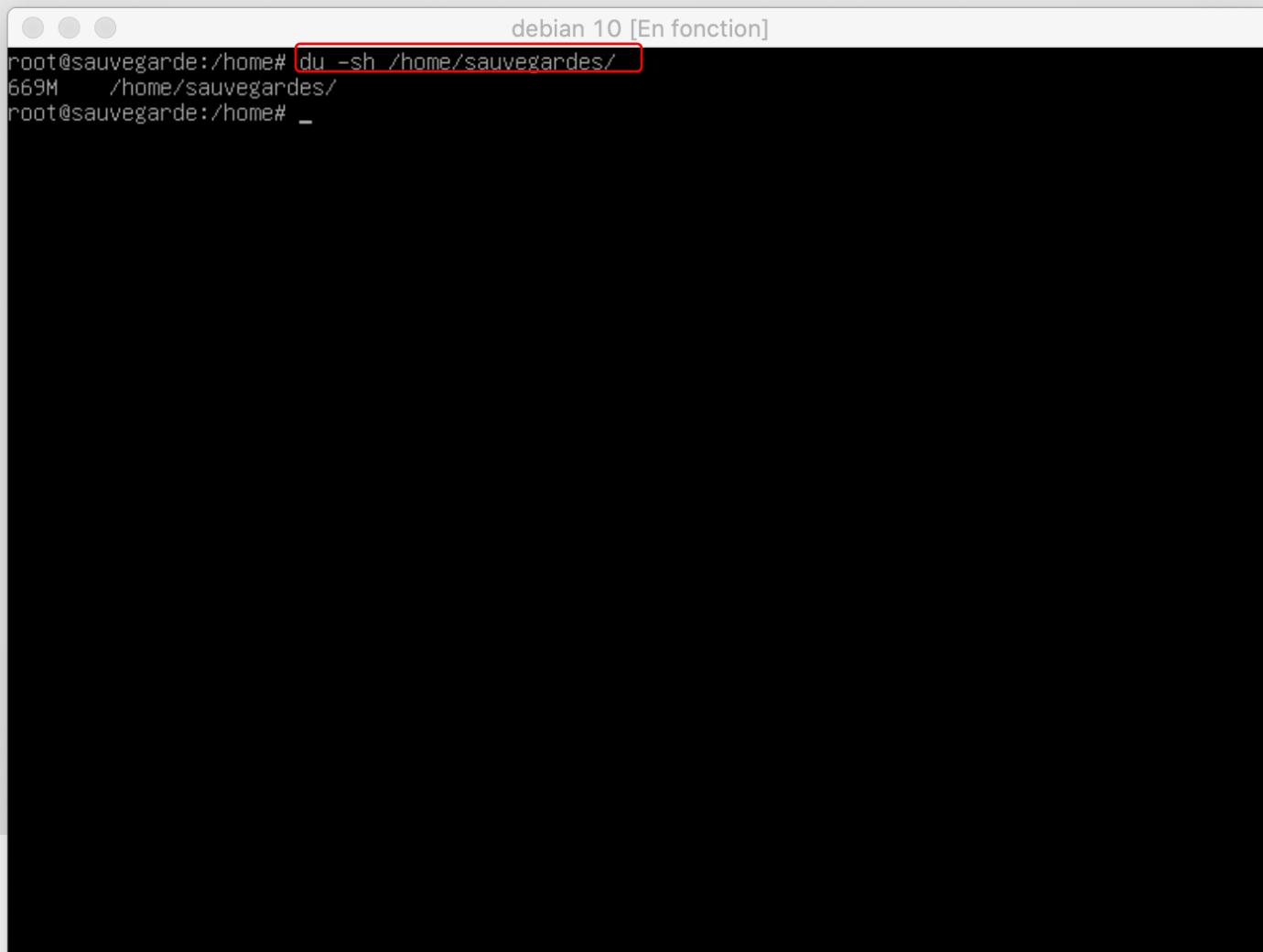
La 2^{ème} ligne permet de faire un test. Le résultat s'affiche dans le fichier.log

Créer un fichier .log est toujours utilisé pour savoir si la commande s'est correctement exécuté.

CONNAÎTRE LA TAILLE DU DOSSIER DE SAUVEGARDE

CONNAÎTRE LA TAILLE DU DOSSIER DE SAUVEGARDE

Il est utile de connaître la taille du dossier de sauvegarde.



The screenshot shows a terminal window titled "debian 10 [En fonction]". The command "du -sh /home/sauvegardes/" is entered, and the output "669M /home/sauvegardes/" is displayed. The command line itself is highlighted with a red box.

```
debian 10 [En fonction]
root@sauvegarde:/home# du -sh /home/sauvegardes/
669M /home/sauvegardes/
root@sauvegarde:/home#
```