

Lab - Montage d'un segment RamDisk

Un RamDisk est un segment de mémoire RAM système agissant comme un système de fichiers.



En quoi un RamDisk est intéressant ?

Conserver un ensemble de données , comme des tables d'une base de données ou un dictionnaire. un RamDisk accélère les recherches des données car l'accès mémoire est plus rapide que l'accès disque dur. Cela pourrait être approprié, par exemple, pour un serveur http ou un serveur de base de données.

Avantage : Accès très rapide au données (temps lecture/écriture).

Inconvénients : Volatile, perte de données au redémarrage ou à l'arrêt du système,
Moins de RAM disponible pour le système.

Objectifs

Améliorer la vitesse d'accès à un serveur web.

Partie 1: Créer une procédure de sauvegarde et de restauration du répertoire racine résidant dans un segment de RAM d'un serveur web.

Partie 2: Test de bon fonctionnement

Ressources requises

Un ordinateur exécutant une VM Ubuntu sous Oracle VirtualBox.

Ouvrez une fenêtre de terminal dans Ubuntu.

Connectez-vous à Ubuntu à l'aide de Vos informations d'identification:

Cliquez sur l'icône du terminal pour ouvrir une fenêtre de terminal.



Partie 1 : Créer une procédure de sauvegarde et de restauration des données d'un segment de RAM

Étape 1 : Installation d'un serveur web (apache2) et ioping pour le test de débit .

```
user@localhost:~$sudo su
user@localhost:~#cd;apt update ; apt install apache2 ioping
root@localhost:~#cp -r /var/www/html /root/
root@localhost:~#tar cvzf html.tar.gz html/
```

Étape 2 : Création du fichier /etc/init.d/restaure.sh

Le fichier `restaure.sh` contiendra toutes les commandes nécessaires à la restauration des données dans le segment RamDisk du serveur web :

GNU/Linux

```
root@localhost:~# nano /etc/init.d/restaure.sh
#Début
mount -t tmpfs -o size=256M tmpfs /var/www # tmpfs est un système de fichier résidant dans la RAM
cp /root/html.tar.gz /var/www/
cd /var/www
tar xvzf html.tar.gz && rm html.tar.gz
#Fin ( sauvegarder le fichier && quitter nano )
```

Ajouter le droit d'exécution au fichier `restaure.sh` :

```
root@localhost:~# chmod u+x /etc/init.d/restaure.sh
```

Tester l'exécution du fichier `restaure.sh`

La procédure d'appel :

```
root@localhost:~# /etc/init.d/restaure.sh
```

Étape 3 : Création du fichier `/etc/init.d/bckup.sh`

Le fichier `bckup.sh` contiendra toutes les commandes nécessaires à la sauvegarde des données stockées dans le segment `RamDisk` :

```
root@localhost:~# nano /etc/init.d/bckup.sh
#Début
DATE=$(date +%H-%M-%d-%m-%y)
cd /var/www/
tar cvzf html.tar.gz html/ && cp html.tar.gz /root/
ln /root/html.tar.gz /root/html.$DATE.tar.gz

#Fin ( sauvegarder && exit )
```

Ajouter le droit d'exécution au fichier `bckup.sh` :

```
root@localhost:~# chmod u+x /etc/init.d/bckup.sh
```

Tester l'exécution du script `bckup.sh`

La procédure d'appel

```
root@localhost:~# /etc/init.d/bckup.sh
```

Partie 2: Test de fonctionnement

Testez l'efficience de la procédure en vérifiant la disponibilité des données de sauvegarde dans le segment `RamDisk` après le démarrage/redémarrage du système.

Test 1

vérifiez que le segment RAM `tmpfs` est bien monté dans `/var/www`

```
root@localhost:~# mount
tmpfs on /var/www type ext4 (rw)
```

Test 2

vérifiez que le répertoire `/var/www/html` est bien présent.

```
root@localhost:~# cd /var/www/
root@localhost:~# ls
html
```

Test 3

Test de débit via la commande ioping

ioping /home

ioping /var/www

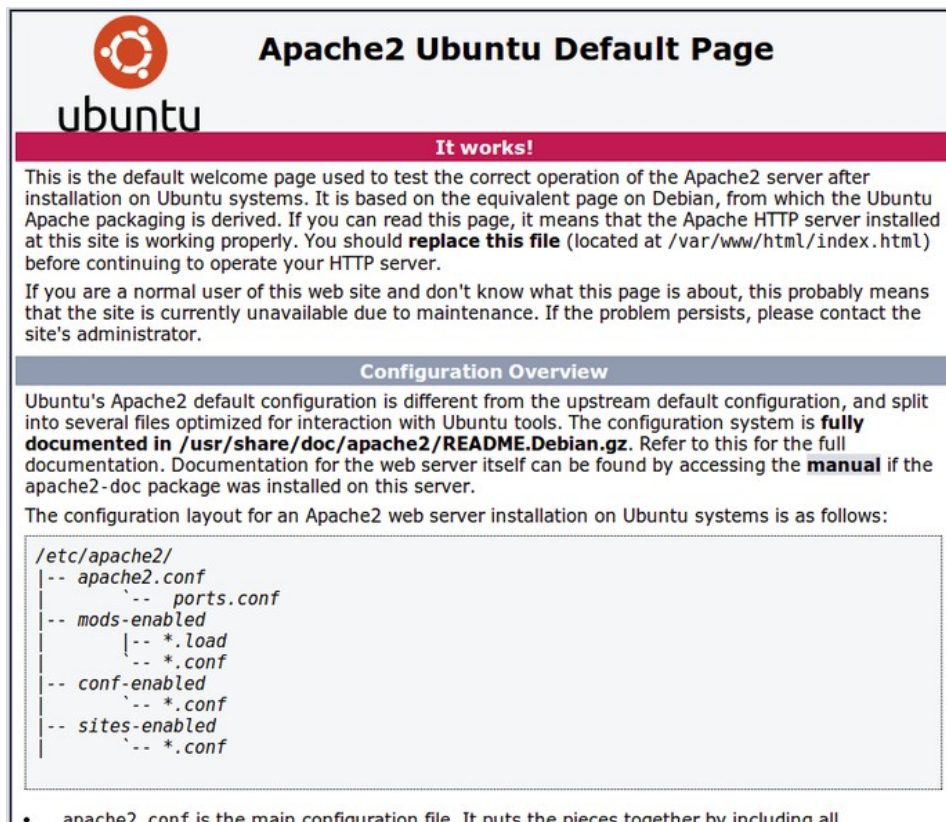
que remarquez-vous ?


Test 4

Test d'accès à la page d'accueil du serveur web

Exécutez le navigateur web firefox et saisissez l'url `http://127.0.0.1`

La page d'accueil d'apache2 devrait s'afficher comme ceci :



 **Apache2 Ubuntu Default Page**

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all