

Étape 3 : Website, Service, Entretien

Dans cette section, nous allons apprendre comment héberger un site web sur la Raspberry Pi pour télécharger les vidéos, puis nous passerons à la création de services pour lancer le code du système 'main.py' au démarrage. Enfin, nous gérerons l'entretien du système en créant des tâches qui s'exécutent régulièrement pour effacer les vidéos sauvegardées.

Website

Pour se Cultiver

Nginx :

Nginx est un serveur web open-source connu pour sa haute performance, sa stabilité, sa faible consommation de ressources et sa capacité à gérer un grand nombre de connexions simultanées. Il est souvent utilisé comme serveur web ou serveur proxy inverse, ainsi que pour la mise en cache, l'équilibrage de charge, et plus encore.

PHP est un langage de script côté serveur conçu pour le développement web, mais il est également utilisé comme langage de programmation généraliste. Pour exécuter du code PHP sur un serveur, vous avez besoin de l'interpréteur PHP installé, tout comme vous avez besoin de l'interpréteur Python pour exécuter du code Python.

PHP-FPM (PHP FastCGI Process Manager) :

PHP-FPM est une implémentation alternative de PHP FastCGI avec des fonctionnalités supplémentaires utiles pour les sites web à forte charge. Il permet à PHP de fonctionner efficacement avec Nginx. Bien que Nginx ne puisse pas directement interpréter PHP, il transmet les requêtes PHP à PHP-FPM, qui traite le code et renvoie le résultat à Nginx.

1-Préparation

1 -Créer un répertoire pour stocker tous les fichiers en rapport avec la structure de votre site, par exemple :

```
mkdir /home/pi/Desktop/website
```

2 -Installer Nginx et PHP

```
sudo apt install nginx
```

```
sudo apt install php8.2-fpm
```

3 -Démarrer et activer Nginx

```
sudo systemctl start nginx
```

```
sudo systemctl enable php8.2-fpm
```

Normalement, après cette étape, vous pouvez visiter votre site en tapant l'adresse IP de votre Raspberry Pi dans votre navigateur web. Vous devriez voir cette page.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

2-Modifier la Configuration de Nginx

Ouvrez le fichier de configuration par défaut pour l'éditer :

```
sudo nano /etc/nginx/sites-available/default
```

Remplacez le contenu du fichier en modifiant `root` avec le chemin du répertoire de votre site web et `alias` avec le chemin du répertoire où vous stockez vos vidéos, par exemple :

```
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    root /home/pi/Desktop/website;  
    index index.php index.html index.htm;  
  
    server_name _;
```

```

location / {
    try_files $uri $uri/ =404;
}

location /videos/ {
    alias /home/pi/Desktop/videos/;
    autoindex on;
    allow all;
}

location ~ \.php$ {
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
}

location ~ /\.ht {
    deny all;
}
}

```

explication :

listen 80 default_server; et listen [::]:80 default_server;;

- Ces directives indiquent à Nginx d'écouter les connexions entrantes sur le port 80, le port par défaut pour HTTP.
- default_server signifie que ce bloc de serveur sera le serveur par défaut si aucun autre bloc de serveur ne correspond à la demande entrante.

index index.php index.html index.htm;;

- Spécifie les fichiers par défaut à servir lorsqu'un répertoire est demandé. Nginx cherchera index.php, index.html, et index.htm dans cet ordre.

server_name _;;

- C'est un nom de serveur générique. Cela signifie que ce bloc de serveur gérera les requêtes qui ne correspondent à aucun autre bloc de serveur avec un nom de serveur spécifique.

location / { ... }:

- Ce bloc configure la manière de gérer les requêtes pour l'URL racine (c'est-à-dire la page d'accueil) et toutes les autres URL non couvertes par d'autres blocs location.

```
try_files $uri $uri/ =404;;
```

- Cette directive tente de servir le fichier spécifié par l'URI (\$uri).
- Si le fichier n'existe pas, il essaie le répertoire (\$uri/).
- Si aucun des deux n'existe, il renvoie une erreur 404.

```
location /videos/ { ... }:
```

- Ce bloc configure la manière de gérer les requêtes pour les URL commençant par /videos/.

```
alias /home/pi/Desktop/videos/;
```

- Définit le répertoire correspondant aux requêtes URL /videos/ à /home/pi/Desktop/videos/.

```
autoindex on;;
```

- Active la liste automatique des répertoires pour le répertoire /videos/ si aucun fichier index n'est trouvé.

```
allow all;;
```

- Autorise l'accès à tous les utilisateurs.

```
location ~ \.php$ { ... }:
```

- Ce bloc gère les requêtes pour les fichiers PHP. La regex ~ \.php\$ correspond à toute requête se terminant par .php.

```
include snippets/fastcgi-php.conf;;
```

- Inclut les paramètres de configuration pour FastCGI, nécessaire pour traiter les fichiers PHP.

```
fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;;
```

- Passe les requêtes PHP au socket Unix PHP-FPM (FastCGI Process Manager) pour traitement.

```
location ~ /\.ht { ... }:
```

- Ce bloc gère les requêtes pour les fichiers commençant par .ht, tels que les fichiers .htaccess.

```
deny all;;
```

- Interdit l'accès à ces fichiers pour des raisons de sécurité, car les fichiers .htaccess sont utilisés pour la configuration et ne doivent pas être accessibles publiquement.

3-Rendre le site plus beau

Créer un fichier pour le style.

```
sudo nano /home/pi/Desktop/website/style.css
```

Ajouter ca :

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  max-width: 600px;
  width: 100%;
}

h1 {
  text-align: center;
  color: #333;
}

ul {
  list-style: none;
  padding: 0;
}

li {
  margin: 10px 0;
}

a {
  text-decoration: none;
```

```
color: #007bff;
font-weight: bold;
}

a:hover {
color: #0056b3;
}
```

en plus vous pouvez ajouter une image à votre site, sauvegardez la dans le répertoire de votre site.

4-Ajoutez un fichier PHP dans le répertoire de votre site web

Créer le fichier index.php

```
sudo nano /home/pi/Desktop/website/index.php
```

ensuite ajouter ca :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Video Downloads</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <h1>Video Downloads</h1>
    
    <ul>
      <?php
        $dir = '/home/pi/Desktop/videos/';
        $webPath = '/videos/';

        if ($handle = opendir($dir)) {
          while (false !== ($file = readdir($handle))) {
            if ($file != "." && $file != ".." && !is_dir($dir . $file)) {
              $filePath = $webPath . $file;
              echo "<li><a href='" . htmlspecialchars($filePath) . "' download>" . htmlspecialchars($file) .
```

```

"</a></li>";
    }
}
closedir($handle);
} else {
    echo "<p>Failed to open directory: $dir</p>";
}
?>
</ul>
</div>
</body>
</html>

```

explication :

Le document commence par une déclaration `<!DOCTYPE html>`, indiquant qu'il s'agit d'un document HTML5.

La balise `<html>` définit la langue comme étant l'anglais avec `lang="en"`.

La section `<head>` contient des métadonnées et des liens, notamment :

- `<meta charset="UTF-8">` définit l'encodage des caractères à UTF-8.
- `<title>` définit le titre de la page à "Video Downloads".
- `<link rel="stylesheet" href="style.css">` lie un fichier CSS externe nommé style.css.

La section `<body>` contient le contenu principal de la page :

- Une div avec la classe container encapsule le contenu.
- Une balise `<h1>` avec le titre "Video Downloads".
- Une balise `` affichant une image (images.png) avec une largeur maximale de 100%.
- Une liste non ordonnée (``) qui sera peuplée dynamiquement avec des éléments de liste (``) par PHP.

Le script PHP est intégré dans le HTML en utilisant les balises `<?php ... ?>`.

`$dir` est défini comme le chemin du répertoire contenant les vidéos (/home/pi/Desktop/videos/).

`$webPath` est défini comme le chemin URL correspondant au répertoire des vidéos (/videos/).

`opendir($dir)` tente d'ouvrir le répertoire spécifié par `$dir`. Si cela réussit, il retourne un handle de répertoire (`$handle`).

`readdir($handle)` lit les entrées du handle de répertoire. Cette boucle continue jusqu'à ce que toutes les entrées aient été lues.

Cette condition assure que les entrées spéciales (. et ..) et les sous-répertoires sont exclus, ne laissant que les fichiers réguliers.

`$filePath` combine le chemin web avec le nom de fichier pour créer l'URL du fichier.

`htmlspecialchars($filePath)` garantit que le chemin du fichier est encodé en toute sécurité pour la sortie HTML.

`htmlspecialchars($file)` garantit que le nom du fichier est encodé en toute sécurité pour la sortie HTML.

Un élément `` est créé avec une balise `<a>`. L'attribut `download` incite le navigateur à télécharger le fichier plutôt qu'à l'ouvrir.

`closedir($handle)` ferme le handle de répertoire.

Si `opendir($dir)` échoue, un message d'erreur est affiché.

5-Modifier les Permissions

```
sudo chown -R www-data:www-data /home/pi/Desktop/website
sudo chown -R www-data:www-data /home/pi/Desktop/videos
sudo chmod -R 755 /home/pi/Desktop/website
sudo chmod -R 755 /home/pi/Desktop/videos
```

Changement de propriétaire : Les commandes `chown` s'assurent que l'utilisateur `www-data` (qui est typiquement l'utilisateur sous lequel le serveur web fonctionne) possède les répertoires et leur contenu. C'est important car cela permet au serveur web de lire et d'exécuter les fichiers selon les besoins.

Changement des permissions : Les commandes `chmod` définissent les permissions afin que le propriétaire (`www-data`) ait un contrôle total (lecture, écriture, exécution), tandis que le groupe et les autres peuvent lire et exécuter les fichiers, mais pas les modifier. Cela permet de sécuriser les répertoires tout en autorisant l'accès nécessaire pour servir les fichiers via le serveur web.

6-Lancer votre site

Redémarrer Nginx et PHP-FPM

```
sudo systemctl reload nginx
```

Maintenant vous pouvez visiter votre site en tapant l'adresse IP de votre Raspberry Pi dans votre navigateur web.

7-Guide de Dépannage

Je ne suis pas un expert en développement web, mais j'ai fait de mon mieux pour vous expliquer comment tout cela fonctionne.

Voici un petit guide de dépannage pour vous aider à résoudre vos problèmes.

Vérification des Permissions

Assurez-vous que l'utilisateur du serveur web (www-data) possède les droits appropriés sur les répertoires et fichiers concernés :

```
sudo ls -l /home/pi/Desktop/website
sudo ls -l /home/pi/Desktop/videos
sudo ls -l /path/to/other/directories
```

Vérification des Logs d'Erreurs Nginx

Vérifiez le journal des erreurs de Nginx pour toute erreur liée à l'accès aux fichiers ou aux permissions :

```
sudo tail -f /var/log/nginx/error.log
```

Comportement du Navigateur

Consultez la Console du Navigateur :

Utilisez les outils de développement du navigateur (F12) pour vérifier les erreurs dans la console liées à l'accès aux fichiers ou aux permissions.

Gestion des Services au Démarrage

Liste des Services Configurés pour Démarrer

Utilisez `systemctl` pour voir quels services sont configurés pour démarrer au boot :

```
systemctl list-unit-files --type=service | grep enabled
```

Vérification des Services en Cours d'Exécution

Vérifiez l'état d'un service spécifique :

```
systemctl status nginx
systemctl status php8.2-fpm
```

Vérification des Logs Détaillés

Consultez les journaux détaillés pour un service spécifique en cas de problème :

```
sudo journalctl -xeu nginx.service  
sudo journalctl -xeu php-fpm.service
```

Validation de la Configuration Nginx

Assurez-vous que la configuration Nginx est correcte en utilisant la commande de test :

```
nginx -t
```

Service

Dans cette partie, nous allons voir comment exécuter le code de notre système `main.py` (de la section précédente) au démarrage.

Pour ce Cultiver

Un fichier de service est un fichier de configuration utilisé par le gestionnaire de systèmes et de services systemd pour définir et gérer les services sous Linux. Ce fichier contient des informations sur la manière dont un service particulier doit être démarré, arrêté et géré par systemd. Il définit les commandes à exécuter pour démarrer le service, les dépendances du service, ainsi que d'autres paramètres qui contrôlent le comportement du service.

Comparaison entre systemd et crontab

Systemd et crontab sont deux outils utilisés pour programmer et gérer des tâches sur un système Linux, mais ils sont utilisés dans des contextes différents et ont des forces et des objectifs distincts. Systemd est conçu pour la gestion des services (processus de longue durée), garantissant leur démarrage au boot, leur redémarrage en cas d'échec, et leur arrêt en douceur. Il gère des dépendances complexes entre services et offre une journalisation centralisée avec journalctl. Il est également capable de redémarrer automatiquement les services en cas de défaillance et de surveiller leur statut. Systemd fournit une interface standardisée sur diverses distributions Linux, ce qui le rend idéal pour la gestion des services en arrière-plan tels que les serveurs web et les serveurs de bases de données, ainsi que pour les scripts critiques nécessitant un fonctionnement continu. Par exemple, exécuter un script Python en tant que service continu qui doit redémarrer en cas de panne est un cas d'utilisation typique pour systemd.

En revanche, crontab est idéal pour la planification des tâches à des moments spécifiques, des intervalles ou des dates. Il est simple et direct à configurer pour des tâches périodiques et peut exécuter des tâches aussi fréquemment que chaque minute. Chaque utilisateur peut avoir son propre crontab, permettant une planification des tâches au niveau de l'utilisateur. Crontab est donc approprié pour les tâches de maintenance périodiques comme les sauvegardes et les nettoyages, ainsi que pour les tâches uniques ou récurrentes comme l'envoi d'e-mails quotidiens ou la génération de rapports. Par exemple, exécuter un script chaque jour à minuit pour sauvegarder

une base de données est une tâche typique pour crontab.

1-Créer un fichier de service Systemd

Créez un fichier de service pour gérer votre script Python. Enregistrez-le dans `/etc/systemd/system/` avec une extension `.service`, par exemple, `camera_system.service`.

```
sudo nano /etc/systemd/system/camera_system.service
```

2-Modifier le fichier de service

Dans l'éditeur ``nano``, ajoutez le contenu suivant au fichier de service :

```
[Unit]
Description=Camera System Service
After=local-fs.target

[Service]
User=root
Group=root
WorkingDirectory=/home/pi/Desktop/scripts/os5
Environment="PATH=/usr/bin:/usr/local/bin:/usr/sbin:/home/pi/Desktop/scripts/venv/bin"
ExecStart=/bin/bash -c ' . /home/pi/Desktop/scripts/venv/bin/activate && python main.py'
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Explication :

- **[Unit]** :

- ``Description`` : Une brève description du service.
- ``After`` : S'assure que le service démarre après que le système de fichiers local soit disponible.

- **[Service]** :

- ``User`` : Exécute le service en tant qu'utilisateur ``root``.
- ``Group`` : Exécute le service dans le groupe ``root``.
- ``WorkingDirectory`` : Définit le répertoire de travail pour le script.
- ``Environment`` : Définit le chemin pour inclure les répertoires nécessaires, y compris l'environnement virtuel Python.

- ``ExecStart`` : Spécifie la commande pour exécuter votre script Python.
- ``Restart`` : Redémarre automatiquement le service s'il se termine avec un statut non nul.
- ``RestartSec`` : Attend 10 secondes avant de redémarrer le service.

- **[Install]** :

- `WantedBy` : S'assure que le service démarre en mode multi-utilisateur (démarrage normal).

3-Vérifiez les Permissions et la Propriété des Fichiers

Exécutez la commande suivante pour vérifier les permissions et la propriété de votre script Python :

```
ls -l /home/pi/Desktop/scripts/os5/main.py
```

Exemple de Sortie :

```
-rwxr-xr-x 1 pi pi 1234 Jun 27 12:34 /home/pi/Desktop/scripts/os5/main.py
```

Explication de la Sortie :

- `rwxr-xr-x` : Ce sont les permissions pour le fichier.
- `rwx` : Le propriétaire (`pi`) a les permissions de lecture, écriture et exécution.
- `r-x` : Le groupe (`pi`) a les permissions de lecture et exécution.
- `r-x` : Les autres ont les permissions de lecture et exécution.
- `1` : Le nombre de liens physiques vers le fichier.
- `pi` : Le propriétaire du fichier.
- `pi` : Le groupe du fichier.
- `1234` : La taille du fichier en octets.
- `Jun 27 12:34` : La date et l'heure de la dernière modification du fichier.
- `/home/pi/Desktop/scripts/os5/main.py` : Le chemin vers le fichier.

Veuillez vérifier les permissions et la propriété en utilisant la commande ci-dessus et confirmer que l'utilisateur `pi` a les permissions nécessaires.

4- Recharger la configuration de Systemd

Rechargez la configuration du gestionnaire Systemd pour reconnaître le nouveau fichier de service et toutes les modifications qui y ont été apportées.

```
sudo systemctl daemon-reload
```

5- Activer et démarrer le service

Activez le service systemd pour qu'il démarre automatiquement au démarrage.

```
sudo systemctl enable camera_system.service
```

Démarrez le service systemd pour lancer votre script Python.

```
sudo systemctl start camera_system.service
```

6-Guide de Dépannage

Arrêter le service

```
sudo systemctl stop camera_system.service
```

Désactiver le démarrage automatique

```
sudo systemctl disable camera_system.service
```

Vérifiez l'état du service.

```
sudo systemctl status camera_system.service
```

Vérifiez les journaux du service :

```
sudo journalctl -u camera.service
```

Entretien

On va utiliser crontab pour entretenir notre système et automatiser la tâche de suppression de vidéos.

1-Ouvrez Crontab pour l'édition

```
crontab -e
```

2-Ajoutez une tâche Cron pour supprimer les vidéos

ajoutez cette ligne :

```
0 2 * * * find /home/pi/Desktop/videos -name "*.avi" -type f -exec rm {} \; >>  
/home/pi/Desktop/videos/deletion.log 2>&1
```

puis sauvegardez et quittez

Explication de l'emploi de temps Cron (0 2 * * *) :

0 : Minutes (0-59)
2 : Heure (0-23)
* : Jour du mois (1-31)
* : Mois (1-12)
* : Jour de la semaine (0-6, où 0 est dimanche)

Ce planning Cron (0 2 * * *) exécute la commande find tous les jours à 2h00 du matin (0 2) pour trouver et supprimer (-delete) les fichiers (-type f) correspondant à *.avi .

find /home/pi/Desktop/videos -name "*.avi" -type f -exec rm {} \; : Recherche tous les fichiers *.avi dans le répertoire /home/pi/Desktop/videos et les supprime (-exec rm {} \;).

/home/pi/Desktop/videos/deletion.log : Redirige la sortie standard (stdout) de la commande find (qui inclut les messages sur les fichiers supprimés) et l'ajoute au fichier deletion.log.

2>&1 : Redirige les erreurs standard (stderr) vers stdout, de sorte que stdout et stderr sont capturés dans le fichier journal.

3-Créez le fichier deletion.log pour enregistrer les messages de suppression.

```
touch /home/pi/Desktop/videos/deletion.log
```

4- Ajuster les permissions du fichier

Définissez les permissions appropriées pour que l'utilisateur exécutant la tâche cron puisse écrire dans le fichier journal. Par exemple, si votre utilisateur est pi :

```
chmod 664 /home/pi/Desktop/videos/deletion.log  
chown pi:pi /home/pi/Desktop/videos/deletion.log
```

chmod 664 : Accorde les permissions de lecture et d'écriture à l'utilisateur et au groupe, et la permission de lecture aux autres.

chown pi : Change la propriété du fichier à l'utilisateur pi et au groupe pi.

5-Vérifier les permissions

Vérifiez les permissions et la propriété du fichier journal.

```
ls -l /home/pi/Desktop/videos/deletion.log
```

Assurez-vous que les permissions affichent (rw-rw-r--) et que la propriété est correcte (pi ou votre utilisateur et votre groupe).

6-Vérifier la tâche Cron

```
crontab -l
```

References :

Host a website on your raspberry Pi : <https://www.youtube.com/watch?v=Y1mNeWwj8D0>

Run a program on your raspberry Pi : <https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/>

FIN

Revision #15

Created 30 June 2024 19:40:02 by El Ghouli Ali

Updated 5 July 2024 12:55:44 by El Ghouli Ali