

Rapport laboratoire #4

ADS - Administration système

14.04.2024

Alexis Martins, Pablo Saez et Thomas Germano

HEIG
ADS

Table des matières

1. Introduction	3
2. Tâche 2 : Création de miniatures	3
3. Tâche 3 : Génération de fichier HTML	4
4. Tâche 4 : Utilisation d'un tunnel SSH	5
5. Conclusion	6

1. Introduction

Ce laboratoire a pour objectif de nous aider à nous familiariser avec les bases du scripting en bash. Il propose plusieurs exercices dans lesquels nous allons devoir utiliser diverses commandes successivement afin de réaliser des tâches précises. Il propose aussi une petite introduction au tunneling avec SSH.

2. Tâche 2 : Création de miniatures

Aperçu du script pour afficher les dimensions :

```
#!/bin/bash

for file in ./lab04_raw_files/*.{png,jpg}
do
    identify -format "$file width: %w, height: %h \n" "$file"
done
```

Aperçu du script pour renommer les images :

```
#!/bin/bash

# Définition du dossier contenant les fichiers originaux
SOURCE_DIR="./lab04_raw_files"
# Définition du dossier de destination pour les fichiers renommés
RENAMED_DIR="$SOURCE_DIR/lab04_renamed_pictures"

# Crée le dossier de destination s'il n'existe pas
mkdir -p "$RENAMED_DIR"

for file in $SOURCE_DIR/*.{png,jpg}
do
    # Extraction des informations nécessaires du chemin du fichier
    suffix=$(identify -format '%w_%h' "$file")
    file_name=$(basename "$file")
    name=${file_name%.*}
    extension=${file_name##*.}

    # Copie le fichier avec le nouveau nom dans le dossier de destination
    cp "$file" "$RENAMED_DIR/$name$suffix.$extension"
done
```

Réponse à la question : Nous avons décidé de simplement créer un dossier qui va contenir les résultats. Ainsi les images se copient là-bas et le script n'y a pas accès, donc il ne fait pas le travail à double.

Aperçu du script pour faire les miniatures des documents :

```
#!/bin/bash

# Définition du dossier contenant les fichiers originaux
SOURCE_DIR="./lab04_raw_files"
# Définition du dossier de destination pour les miniatures
THUMBNAILS_DIR="$SOURCE_DIR/lab04_thumbnails"

# Crée le dossier de destination s'il n'existe pas
mkdir -p "$THUMBNAILS_DIR"

for file in $SOURCE_DIR/*.{png,jpg,pdf}
do
    # Extraction des informations nécessaires du chemin du fichier
```

```

file_name=$(basename "$file")
name=${file_name%.*}
extension=${file_name##*.}
path=$(dirname "$file")

# Commande pour créer la miniature
if [ "$extension" = "pdf" ]
then
    thumbnail_file="$THUMBNAIIS_DIR/${name}_thumb.jpg"
    convert -geometry 300 "$file"[0] "$thumbnail_file"
else
    thumbnail_file="$THUMBNAIIS_DIR/${name}_thumb.$extension"
    convert -geometry 300 "$file" "$thumbnail_file"
fi
done

```

Réponse à la question : Comme pour le script précédent, nous avons juste créer un dossier qui va contenir toutes les miniatures produites

3. Tâche 3 : Génération de fichier HTML

Nous sommes partis sur la version HTML plutôt que la version PHP.

```

#!/bin/bash

# Définir les chemins
RAW_FILES_DIR="./raw_files/lab04_raw_files"
THUMBNAIIS_DIR="./raw_files/lab04_thumbnails"
DEST_DIR="."
BEGIN_TEMPLATE="${DEST_DIR}/template_begin.html"
END_TEMPLATE="${DEST_DIR}/template_end.html"
PAGE="${DEST_DIR}/page.html"

# Créer ou vider le fichier page.html
> "${PAGE}"

# Ajouter le début du template
cat "${BEGIN_TEMPLATE}" > "${PAGE}"

# Section des images
cat <<EOF >> "${PAGE}"
<article class="container article">
    <div class="row">
        <div class="col-md-10 col-md-pull-3 col-md-offset-4 article__content">
            <div><div><h2>Découvrez-nous en images</h2></div></div>
            <div class="row">
EOF

# Générer les entrées pour les images
for img in ${RAW_FILES_DIR}/*.png,jpg; do
    filename=$(basename "$img")
    thumb="${THUMBNAIIS_DIR}/${filename%.*}_thumb.${filename##*.}"
    if [[ -f "$thumb" ]]; then
        echo "<div class=\"col-md-6 col-xs-12\"><a href=\"\$img\"><img class=\"vignette\"
src=\"\$thumb\" /></a></div>" >> "${PAGE}"
    fi
done

```

```
# Fin de la section des images et début de la section des brochures
cat <<EOF >> "${PAGE}"
    </div>
    </div>
</div>
<div class="row" style="margin-top: 40px;">
    <div class="col-md-10 col-md-pull-3 col-md-offset-4 article__content">
        <div><div><h2>Téléchargez nos brochures</h2></div></div>
        <div class="row">
EOF

# Générer les entrées pour les brochures
for pdf in ${RAW_FILES_DIR}/*.pdf; do
    filename=$(basename "$pdf")
    thumb="${THUMBNAILS_DIR}/${filename%.*}_thumb.jpg"
    if [[ -f "$thumb" ]]; then
        echo "<div class=\"col-md-6 col-xs-12\"><a href=\"\${pdf}\"><img class=\"vignette\"
src=\"\${thumb}\" /></a></div>" >> "${PAGE}"
    fi
done

# Fin de la section dynamique
cat <<EOF >> "${PAGE}"
    </div>
    </div>
</div>
</article>
EOF

# Ajouter la fin du template
cat "${END_TEMPLATE}" >> "${PAGE}"
```

4. Tâche 4 : Utilisation d'un tunnel SSH

La commande utilisée pour créer le tunnel est la suivante :

```
#!/bin/bash
```

```
ssh -L 1337:localhost:3306 labh@ads.iict.ch
```

Une fois le tunnel effectué, il était possible de se connecter avec Workbench à la base de données.

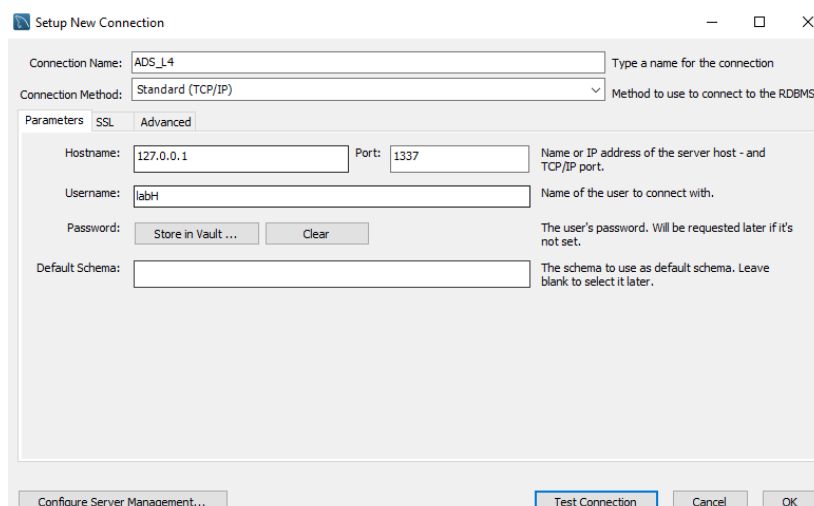


Fig. 1 — Connexion

On a réussi à correctement consulter les informations présentes dans la base de données.

The screenshot shows a database management interface. On the left, a 'SCHEMAS' pane displays a tree view of the database structure: 'labH' contains 'Tables', which contains 'menu'. The 'menu' table has columns: 'id', 'name', 'level', and 'url'. Below the columns, there are sections for 'Indexes', 'Foreign Keys', 'Triggers', 'Views', 'Stored Procedures', and 'Functions'. The main pane shows a query: 'SELECT * FROM labH.menu;'. Below the query, a 'Result Grid' displays the data from the 'menu' table.

id	name	level	url
1	A propos	1	https://heig-vd.ch/a-propos
2	HEIG-VD	2	https://heig-vd.ch/a-propos/heig-vd
3	Documents officiels	3	https://heig-vd.ch/a-propos/heig-vd/document...
NULL	NULL	NULL	NULL

Fig. 2 — Contenu de la base de données

Finalement nous avons apportés une petite modification à la table en passant Documents officiels --> Document(s) officiel(s) . Voici le résultat en live sur le serveur.



Fig. 3 — Résultat de la modification

5. Conclusion

Nous avons maintenant correctement appris à réaliser des petits scripts bash afin de réaliser des opérations de la vie quotidienne. On remarque que bash est très puissant et nous n'étions pas tous totalement au courant du niveau de manipulations qu'il était possible d'atteindre uniquement avec du scripting en ligne de commandes.