|  |  |
| --- | --- |
|  | LE PROJET CVManager |
|  | C:\Users\Ftheriault\Desktop\blur-comm.png |
|  | **Création du CVManager** |
|  | Version 2.0 |

|  |  |  |
| --- | --- | --- |
|  |  | Table des matiÈres |

[Table des matiÈres 1](#_Toc482632493)

[Description gÉnÉrale du mandat 2](#_Toc482632494)

[Aperçu du système - CVManager 2](#_Toc482632495)

[Page principale 2](#_Toc482632496)

[Page de gestion du serveur 3](#_Toc482632497)

[Votre mandat 4](#_Toc482632498)

[Étapes à suivre 4](#_Toc482632499)

[Explication des tuiles 5](#_Toc482632500)

[Section « Général » 5](#_Toc482632501)

[Tuile : Identification 5](#_Toc482632502)

[Tuile : Redémarrer 5](#_Toc482632503)

[Tuile : Mise à jour 5](#_Toc482632504)

[Tuile : Ressource 6](#_Toc482632505)

[Tuile : Log 6](#_Toc482632506)

[Tuile : SSH Log 6](#_Toc482632507)

[Tuile : Backup 7](#_Toc482632508)

[Tuile : Virus Check 7](#_Toc482632509)

[Tuile : URL 8](#_Toc482632510)

[Section « Usagers » 9](#_Toc482632511)

[Tuile : Créer 9](#_Toc482632512)

[Tuile : Supprimer 9](#_Toc482632513)

[Tuile : Afficher 10](#_Toc482632514)

[Tuile : Verrouiller 10](#_Toc482632515)

[Tuile : Déverrouiller 10](#_Toc482632516)

[Tuile : Changer MDP 10](#_Toc482632517)

[Autres spécifications 11](#_Toc482632518)

[Apache 11](#_Toc482632519)

[SSH 11](#_Toc482632520)

[SFTP 11](#_Toc482632521)

[Identification des scripts 11](#_Toc482632522)

[Protection contre le brute force attack 11](#_Toc482632523)

[Mise à jour automatique de sécurité 11](#_Toc482632524)

[Grille de correction 12](#_Toc482632525)

[Remise 12](#_Toc482632526)

|  |  |  |
| --- | --- | --- |
|  |  | Description gÉnÉrale du mandat |

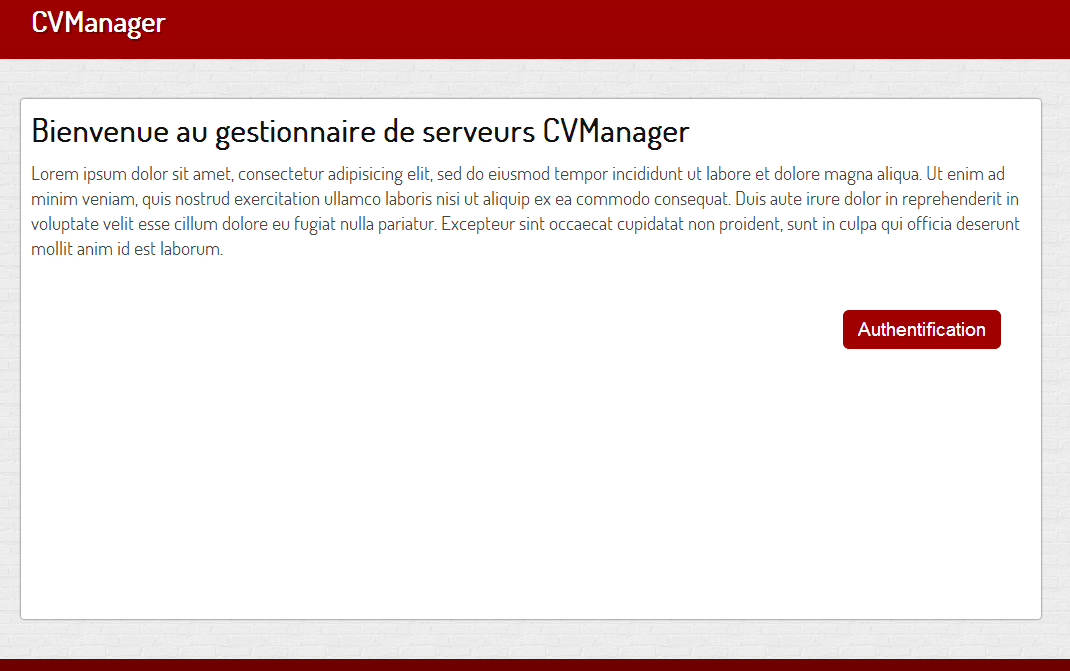
Blur Communications, la plus grande entreprise montréalaise fournissant des services en communications, design et marketing, est présentement en train de développer un système de gestion de serveurs, similaire à CPanel et Plesk.

Cependant, dû à un manque de connaissances à l’interne, ce système, nommé CVManager, n'est toujours pas terminé. Étant vous-même expert en Linux, vous avez été mandaté pour le terminer.

Votre travail consistera donc à terminer le système en y apportant les scripts manquants et en complétant les spécifications demandées. Celui-ci peut se faire seul ou en équipe de 2.

Aperçu du système - CVManager

### Page principale



Cette page contient le formulaire d’authentification du système.

### Page de gestion du serveur

Graphical user interface, application, PowerPoint

Description automatically generated

Cette page contient la liste des tuiles (ou options) du système. Par exemple, la tuile « Créer » permet de créer un usager. Pour ce faire, elle utilise le script : create-user.sh.

Tous les scripts se trouvent dans le dossier Scripts/ du système.

Votre mandat

En analysant ce présent document, vous aurez à installer l’application CVManager dans son état actuel, à modifier la configuration de machines Linux et à faire des scripts Bash.

Étapes à suivre

1. Allez chercher le code existant de CVManager sur notes-de-cours.com et créez votre projet git sur **github** (privé).
2. Faites un clone du projet sur votre ordinateur, et démarrez-le dans Docker (lire le fichier README.md, du projet git).
3. Puisque le dossier « Scripts » est un volume, vous pouvez travailler directement dans votre éditeur de texte, et les modifications se répercuteront automatiquement dans votre *container*.
4. Commencez par travailler sur la tuile « système-authenticate.sh », afin d’être en mesure de vous connecter au système (voir les commentaires dans le fichier pour les détails).
5. Complétez chaque tuile et spécifications du système. **Faire des git commit/pull/push régulièrement !**

* N’oubliez pas de tester chaque tuile!
* ... et… code a little, test a little !

|  |  |  |
| --- | --- | --- |
|  |  | Explication des tuiles |

## Section « Général »

### Tuile : Identification

Argument : 0

Nom du script : identification.sh

Cette tuile sert uniquement à montrer quel usager est utilisé par l’application CVManager. La réponse devrait être « root ».

#!/bin/bash

echo "CVManager utilise l'usager : "

??? (commande qui permet de savoir avec quel usager nous sommes actuellement connectés)

### Tuile : Info. Git

Argument : 0

Nom du script : git.sh



Cette tuile permet d’afficher certaines informations de votre projet Git :

* Le nombre de commits faits jusqu’à présent
* Les informations des 2 derniers commits (voir image)

### Tuile : Ressource

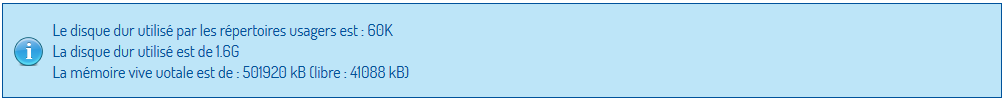
Argument : 0

Nom du script : usage.sh

Cette tuile permet d’afficher 3 informations :

* la taille du disque dur utilisée dans le répertoire /home (le dossier des documents des usagers)
* la taille du disque dur utilisée sur le serveur et
* la mémoire vive totale (et libre !)

Voici un exemple de résultat :



### Tuile : Backup

Argument : 0

Nom du script : backup.sh

Cette tuile permet de faire une copie de sauvegarde du dossier /home (là où il y a les documents importants du serveur) et de la télécharger.

Note :

- Le format du fichier (copie de sauvegarde) doit être **tar.gz** (ex : tar -czvf doc.tar.gz /home/usager)

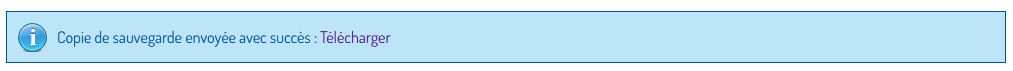
- Cette copie de sauvegarde doit être placée dans le dossier /usr/local/backups (si le dossier n’existe pas, créez-le dans votre script)

- Le nom du fichier doit être la date et l’heure du jour ex : « 2023-05-05 0930.tar.gz »

- Il doit être possible de télécharger la copie de sauvegarde via l’interface Web. Vous devez donc :

1- Faire un lien symbolique (*soft*) du dossier /usr/local/backups vers /var/www/html/Web/web

2- Dans le texte de retour, ajouter un hyperlien avec cet href : /backups/*nom-du-backup*.tar.gz



### Tuile : Virus Check

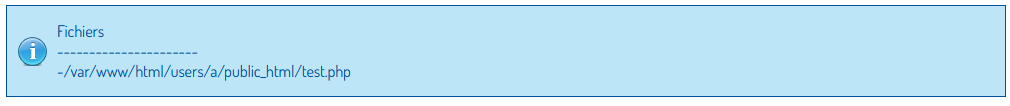
Argument : 0

Nom du script : virus-check.sh

Cette tuile permet de vérifier que les usagers n’essaient pas d’abuser du serveur. Le script de cette tuile doit analyser tous les fichiers de tous les usagers créés par le CVManager (leurs fichiers devraient se trouver sous /home). Si, dans l’un des fichiers, il y a dans la même ligne le terme « shell\_exec » et « su », alors le serveur est corrompu.

Si le serveur n’est pas corrompu, la tuile doit afficher : « Serveur intègre ».

Si le serveur est corrompu, la tuile doit afficher un résultat similaire à ceci :



### Tuile : URL

Argument : 1 (URL valide)

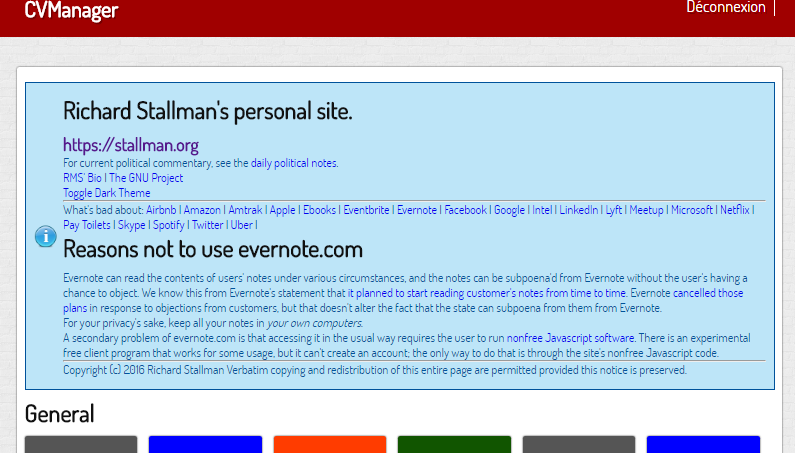
Nom du script : url.sh

Afficher le rendu d’une page Web. La tuile doit aller chercher le contenu (html) de la page Web, puis supprimer l’entête et le pied de page.

Plus précisément, il faut retirer :

* du début du fichier jusqu’à <body>, inclusivement
* la ligne qui contient </body> et celle qui contient </html>

Voici un exemple de résultat pour l’URL : <https://stallman.org/evernote.html>



Le rendu ne sera pas beau…, la fonctionnalité étant limitée, seulement le HTML sera affiché dans la boîte bleue. Il pourrait également y avoir des sites qui ne fonctionnent pas, car cela dépend de la programmation du site Web. Les pages de Richard Stallman fonctionnent bien ;)

## Section « Usagers »

### Tuile : Créer

Argument : 2 (nom et mot de passe)

Nom du script : create-user.sh

Cette tuile doit faire plusieurs choses :

1. Créer un usager
2. Spécifier son mot de passe
3. Modifier son groupe pour : cvmgroup (il faut donc le créer dans le script, s’il n’existe pas)
4. Son répertoire personnel doit être **/home/<nom usager>**

Note : Si aucun mot de passe n’est spécifié dans l’interface du CVManager, ne pas créer l’usager et afficher une erreur similaire à ceci : « Le compte doit avoir un mot de passe ».

Pour activer SSH lorsque votre *container* est en cours, écrivez la commande. Cela vous permettra de vous connecter par SSH à votre *container* (via le port 2200)

docker exec -d projet-website-1 service ssh start

Pour transférer des fichiers via SFTP, vous pouvez utiliser FileZilla, par exemple :



Pour vous connecter en SSH, vous pouvez utiliser la commande suivante (dans git Bash, par exemple) :



### Tuile : Supprimer

Argument : 1 (nom de l’usager à supprimer)

Nom du script : delete-user.sh

En spécifiant le nom de l’usager à supprimer, celui-ci est supprimé du serveur (son répertoire personnel également). Pour que la commande fonctionne, l’usager doit avoir un UID égal ou supérieur à 1000.

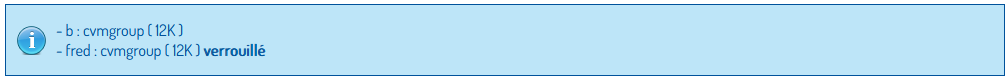
Si la suppression ne peut être réalisée, afficher un message d’erreur (ex : désolé, cet usager ne peut être supprimé).

### Tuile : Afficher

Argument : 0

Nom du script : list-users.sh

Cette tuile permet d’afficher les utilisateurs existants sur le serveur. Le résultat affiche les groupes des usagers et si l’usager est verrouillé. Finalement, la tuile doit afficher la taille son espace personnel utilisé. Voici à quoi devrait ressembler le résultat de la tuile :



Seulement les usagers qui possèdent un UID égal ou supérieur à 1000 sont affichés. L’usager nobody ne doit pas être affiché, ni l’usager que vous utilisez pour vous connecter en ssh, exemple : « sshadmin ».

### Tuile : Verrouiller

Argument : 1 (nom de l’usager à verrouiller)

Nom du script : lock-user.sh

Cette tuile permet de verrouiller un usager. Pour que la commande fonctionne, l’usager doit avoir un UID égal ou supérieur à 1000. Autrement, un message d’erreur est affiché.

### Tuile : Déverrouiller

Argument : 1 (nom de l’usager à déverrouiller)

Nom du script : unlock-user.sh

Cette tuile permet de déverrouiller un usager. Pour que la commande fonctionne, l’usager doit avoir un UID égal ou supérieur à 1000. Autrement, un message d’erreur est affiché.

### Tuile : Changer MDP

Argument : 2 (nom de l’usager et nouveau mot de passe)

Nom du script : update-password.sh

Cette tuile permet de modifier le mot de passe d’un usager. Pour que la commande fonctionne, l’usager doit avoir un UID égal ou supérieur à 1000. Autrement, un message d’erreur est affiché.

|  |  |  |
| --- | --- | --- |
|  |  | Autres spécifications |

### Protection contre le *brute force attack*

En modifiant le script system-authenticate.sh, vous devez vous assurez qu’il n’est pas possible de faire un *brute force attack* pour deviner le mot de passe de CVManager.

Après tout, quelqu’un qui a accès au CVManager peut faire ce qu’il veut avec votre serveur…

Au début du script, écrivez en commentaire la stratégie utilisée pour y parvenir.

Lorsque l'usager ne peut plus se connecter (étant donné son nombre trop important de mauvaises tentatives de connexion), le script ne doit plus retourner zéro (0), mais moins un (-1). Ceci affichera un message différent dans la boîte d’erreur du CVManager. Vous devez programmer votre solution, pas installer un outil déjà créé!

|  |  |  |
| --- | --- | --- |
|  |  | Grille de correction |

|  |  |  |
| --- | --- | --- |
| Programmation (15%) |  | |
| * Utilisation de Git par toute l’équipe (commit+messages et push/pull) * Qualité de la programmation et du travail en général | | 8 % 7 % |
| Tuiles : Général (35%) | |  |
| * Tuile : Identification * Tuile : Info. git * Tuile : Ressources * Tuile : Backup * Tuile : Virus-check * Tuile : URL | | 2 % 6 % 6 %  7 %  8 %  6 % |
| Tuiles : Usagers (35%) | |  |
| * Tuile : Créer * Tuile : Supprimer * Tuile : Afficher * Tuile : Verrouiller * Tuile : Déverrouiller * Tuile : Changer mot de passe | | 5 %  5 %  10 %  5 %  5 %  5 % |
| Sécurité (15%) | |  |
| * Protection du CVManager contre le « brute force attack » | | 15 % |
| Total : | | **100 %** |

|  |  |  |
| --- | --- | --- |
|  |  | Remise |

Le projet doit être remis au maximum lors du dernier cours de la session. Je vous demanderai de me présenter votre projet, et nous testerons les fonctionnalités ensemble. Je regarderai également votre historique de commits, sur github.