

Jeremy Tarrieu - MARE.DATA

Guide utilisation VM

Description de la vm

- 100GB Disque
- 16GB RAM

Users:

- **epf** : accessible depuis l'exterieur
 - **root** : accessible depuis **epf**
 - **desreumaux** : compte du client
-

Connexion au terminal de la vm

Pour accéder au terminal la vm il faut:

- se connecter au compte **epf** par *ssh*
 - pour effectuer des opérations simples telles que créer lire/éditer/exécuter des fichier non protégés et accessibles au compte **epf**.
- se connecter en root via la commande **su**
 - pour effectuer des installations, faire des manipulations sur les containers ect...

```
# connexion au compte epf de la vm via ssh
ssh epf@<adress_ip_vm>
# ... puis entrez le mot de passe du compte epf
```

Vous devriez avoir cette output comme celui-ci:

```
Linux maredata-es 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1
(2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Oct 30 16:23:36 2023 from 81.64.172.100
epf@maredata-es:~$
```

Vous êtes maintenant connecté sur la session epf de la vm.

Vos actions sont limités car cette session ne dispose pas de tous les droits de lecture, d'écriture et d'exécution des fichiers.

Afin d'avoir tous les droits il faut de connecter au compte **root**.

```
su
# ... puis entrez le mot de passe du compte root
```

Vous devriez avoir cela:

```
root@maredata-es:/home/epf#
```

Vous pouvez maintenant travailler !

Conseils de connexion

OPTION 1

Faites comme dit précédemment, cela est long de devoir récupérer le mot de passe.

OPTION 2

Afin d'aller plus vite sur les étapes de connexion il est conseillé de garder en **variable bash** l'adresse IP et les *mot de passe*.

Pour cela il faut créer des variables au sein de notre terminal. Pour cela il faut éditer le fichier `.bashrc` situé dans le *home* de notre session afin d'y ajouter les variables à chaque initialisation du nouveau terminal.

- ouvrez le fichier `~/.bashrc` avec un éditeur de texte (nano, vim).

```
sudo nano ~/.bashrc
```

- ajoutez ces lignes à la fin du fichier. `ctrl+c ctrl+v`

```
# Identifiants de connexion à la vm du projet MARE.DATA EPF
mare_data_epf=<mot_de_passe_epf>
mare_data_root=<mot_de_passe_root>
mare_data_ip=<adresse_ip>
```

- sauvegardez et quittez `ctl+s ctl+x`

Ainsi pour se connecter entrez les commandes suivantes :

```
echo $mारे_data_epf # affiche le mot de passe pour le compte epf
echo $mारे_data_root # affiche le mot de passe pour le compte root
ssh epf@$mारे_data_ip # initialise la connexion ssh sur la vm sur le compte
epf
# ... copiez collez le mot de passe epf et appuyez sur Entrée
su
# ... copiez collez le mot de passe root et appuyez sur Entrée
```

OPTION 3

Voici les étapes de création d'un script bash permettant de se connecter automatiquement à la vm. C'est la méthode la plus rapide. Elle nécessite de s'être déjà connecté une fois via la méthode 1 ou 2 .

- installez sshpass:

```
# sudo
apt install sshpass
```

ssh pass permet de se connecter en ssh sans avoir à entrer le mot de passe de façon interactive, cela permet notamment d'automatiser la connexion ssh via un commande bash.

- créer le script bash:

```
# nous utilisons nano mais vous pouvez utiliser n'importe quel éditeur de
code.
# nano /usr/bin/<nom_de_la_commande_à_entrer_pour_se_connecter>
nano /usr/bin/oob
```

Copiez collez le code suivant en modifiant les mots de passe et l'IP:

```
# changez les mots de passe et l'url
epf="<mot_de_passe_epf>"
root="<mot_de_passe_root>"
ip="<ip_de_la_vm>"

echo "root password : $root" # on affiche le mot de passe afin de pouvoir
se connecter en root
# si jamais les ports viennent à changer vous pourrez les changer ici
jupyter="8888"
phpMA="8080"
kibana="5601"
elastic="9200"

# on imprime les urls dans la console afin de pouvoir cliquer dessus
echo "Jupyter          : http://localhost:$jupyter/"
echo "php-my-admin      : http://localhost:$phpMA/"
```

```

echo "Kibana           : http://localhost:$kibana/"
echo "Elastic          : http://localhost:$elastic/"

# connexion ssh en exposant les ports
sshpass -p $epf ssh \
  -L "$jupyter:$ip:$jupyter" \
  -L "$phpMA:$ip:$phpMA" \
  -L "$kibana:$ip:$kibana" \
  -L "$elastic:$ip:$elastic" \
  "epf@$ip"

```

Appuyez sur **ctrl+S** puis **ctrl+X** afin d'enregistrer et quitter nano.

- *FACULTATIF* - ajouter la chemin du code à la variable d'environnement **\$PATH** dans le fichier **.bashrc** afin de pouvoir appeler le script avec son nom:

```
echo 'export PATH="/usr/bin/oob:$PATH"'
```

- rendre le script executable:

```
# chmod 700 /usr/bin/<nom_de_la_commande_à_entrer_pour_se_connecter>
chmod 700 /usr/bin/oob
```

- se connecter :

```
# <nom_de_la_commande_à_entrer_pour_se_connecter>
oob
```

Ce qui donne cette output :

```

<mot_de_passe_root>
Jupyter           : http://localhost:8888/
php-my-admin      : http://localhost:8080/
Kibana            : http://localhost:5601/
Elastic           : http://localhost:9200/
Linux maredata-es 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1
(2023-09-29) x86_64

```

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

```
Last login: Fri Nov  3 19:05:44 2023 from 37.66.212.87
epf@maredata-es:~$
```

Accès aux ports

Dans le cadre du projet nous avons besoin d'accéder à certaines interfaces web accessibles depuis certains ports de la vm. La connexion ssh ne dispose pas d'interface graphique pour utiliser ces applications.

Pour accéder à ces ports, nous allons les rediriger sur les ports de notre machine personnelle toujours avec la commande **ssh** avec l'argument **-L** qui permet de faire du **port forwarding** ou **port mapping**. (On peut appeler l'argument **-L** autant de fois qu'on le souhaite à condition que les ports soient accessibles)

```
ssh -L <port_vm>:<ip_vm>:<port_perso> <user_vm>@<ip_vm>
```

Dans notre cas de figure les ports à rediriger sont les suivants:

- 8080 - PhpMyAdmin
- 8888 - JupyterHub
- 9200 - ElasticSearch
- 5601 - Kibana

Pour forwarder les ports, se connecter au compte **epf** suffit :

```
echo $mare_data_epf
ssh -L 8080:$mare_data_ip:8080 \
    -L 8888:$mare_data_ip:8888 \
    -L 9200:$mare_data_ip:9200 \
    -L 5601:$mare_data_ip:5601 \
    epf@$mare_data_ip
# ... copiez collez le mot de passe epf et appuyez sur Entrée
```

Sur votre navigateur web:

- http://localhost:8080 phpmyadmin
- http://localhost:8888 jupyter
- http://localhost:9200 elasticsearch
- http://localhost:5601 kibana

Voilà ! Les application web de notre vm sont maintenant accessibles sur notre machine personnelle.

Dossiers et fichiers de la vm

L'architecture de fichiers ressemble à cela :

```
/home/epf/  
.  
├─ docker_install.sh  
└─ project  
   └─ docker-compose.yaml  
      └─ src  
         └─ batch  
            └─ streaming
```

docker_install.sh

Ce fichier executable contient toutes les commandes pour installer docker engine sur la machine.

docker-compose.yaml

Ce fichier contient la configuration de tous les conteneurs lancés par docker.

src/

Ce dossier partage un volume avec le conteneur jupyter. Il contient tous les scripts que l'on peut développer dans le conteneur jupyter. Une grande partie des livrables de code seront stockés dans ce dossier.

Comment éditer des script

Pour éditer les scripts de batch, de streaming ect... il faut :

- se connecter à la vm en exposant au moins le port de jupyter

```
echo $mare_data_epf  
ssh -L 8888:$mare_data_ip:8888 epf@$mare_data_ip  
# ... copier coller le mot de passe epf
```

- ouvrir jupyter sur votre navigateur <http://localhost:8888/>
- éditer ...

php My Admin

Pour se connecter à php My Admin

field	value
host	maria:3306
user	root
password	admin

au sein d'un même **network** docker, les adresses IP des différents conteneurs peuvent être remplacées par le nom du container. D'où l'host **maria:3306**.

Si cela ne marche pas vous pouvez toujours vérifier les valeurs des username et password dans le fichier docker-compose.yaml dans la section maria.

Pour le moment il est conseillé de se connecter en root pour ajouter des tables ect... Bien qu'un user maria data soit déjà présent il n'as pas encore de droits d'accès en particulier.