

Projets CIS

Préambule

- **Chaque projet doit impérativement être accompagné de sa déclaration de traitement RGPD validée par le DPO**

* *Changement de Nom en CIS-LAB*

Logiciels installés post-installation

* *Base de données*

```
cis-lab:~$ apt install nginx mariadb-server
```

```
cis-lab:~$
```

```
root@cis-lab: ~ # mysql_secure_installation
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
haven't set the root password yet, you should just press enter here.
```

```
Enter current password for root (enter for none):
```

```
OK, successfully used password, moving on...
```

```
Setting the root password or using the unix_socket ensures that nobody  
can log into the MariaDB root user without the proper authorisation.
```

```
You already have your root account protected, so you can safely answer 'n'.
```

```
Switch to unix_socket authentication [Y/n] n
```

```
... skipping.
```

```
You already have your root account protected, so you can safely answer 'n'.
```

```
Change the root password? [Y/n] Y
```

```
New password:
```

```
Re-enter new password:
```

```
Password updated successfully!
```

```
Reloading privilege tables..
```

```
... Success!
```

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

```
Remove anonymous users? [Y/n] Y
... Success!
```

Normally, root should only be allowed to connect from 'localhost'. This ensures that someone cannot guess at the root password from the network.

```
Disallow root login remotely? [Y/n] Y
... Success!
```

By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.

```
Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

Reloading the privilege tables will ensure that all changes made so far will take effect immediately.

```
Reload privilege tables now? [Y/n] Y
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

* *Contener Docker* <https://docs.docker.com/engine/install/debian/#install-using-the-repository> 1-Set up Docker's apt repository.

```
# Add Docker's official GPG key:
$ sudo apt-get update
$ sudo apt-get install ca-certificates curl
# $ sudo install -m 0755 -d /etc/apt/keyrings #
$ sudo curl -fsSL https://download.docker.com/linux/debian/gpg -o
/etc/apt/keyrings/docker.asc
$ sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/debian \  
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

2-Install the Docker packages. To install the latest version, run:

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-  
plugin docker-compose-plugin
```

3-Verify that the installation is successful by running the hello-world image:

```
$ sudo docker run hello-world  
Hello from Docker!  
This message shows that your installation appears to be working correctly.
```

- *inotify-tools*

```
cis-lab:~$ apt install inotify-tools
```

From:
<https://portail.emse.fr/dokuwiki/> - **DOC**

Permanent link:
<https://portail.emse.fr/dokuwiki/doku.php?id=recherche:cis-projets&rev=1710237661>

Last update: **12/03/2024 11:01**

