Documentation technique

BTS Services informatique aux organisations

Option Solutions d'Infrastructure, Systèmes et Réseaux (S.I.S.R)

SITUATION N°1

Nom de l'étudiant : LOPEZ-SIGURA Florian | FLS

Auteur	Date	Description
FLS	23/04/2025	Création du document

Table des matières :

Time zone :

Faire la commande “ TZSELECT “ pour pouvoir avoir le bon fuseau horaire local :

```
root@intralabFLS:/home/sio# tzselect
Please identify a location so that time zone rules can be set correctly.
Please select a continent, ocean, "coord", or "TZ".
1) Africa                                                       7) Europe
2) Americas                                                     8) Indian Ocean
3) Antarctica                                                   9) Pacific Ocean
4) Asia                                                        10) coord - I want to use geographical coordinates.
5) Atlantic Ocean                                              11) TZ - I want to specify the timezone using the Posix TZ format.
6) Australia
#? 8
Please select a country whose clocks agree with yours.
1) British Indian Ocean Territory
2) French S. Terr.
3) Maldives
4) Mauritius
#? 4

The following information has been given:

        Mauritius

Therefore TZ='Indian/Mauritius' will be used.
Selected time is now:    Sat May 10 23:30:01 +04 2025.
Universal Time is now:  Sat May 10 19:30:01 UTC 2025.
Is the above information OK?
1) Yes
2) No
#? y
Please enter a number in range.
#? 1

You can make this change permanent for yourself by appending the line
        TZ='Indian/Mauritius'; export TZ
to the file '.profile' in your home directory; then log out and log in again.

Here is that TZ value again, this time on standard output so that you
can use the /usr/bin/tzselect command in shell scripts:
Indian/Mauritius
root@intralabFLS:/home/sio# date
sam. 10 mai 2025 23:30:49 +04
root@intralabFLS:/home/sio#
```

Pour évité de tapper /sbin/ à chaque fois

nano ~/.bashrc

export PATH=$PATH:/sbin:/usr/sbin

source ~/.bashrc

Mode opératoire MariaDB

Étape 1 : Connexion à MariaDB

```
mysql -u root -p
```

Étape 2 : Création de la base de données

```
CREATE DATABASE gsb_frais CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
Étape 3 : Création de l'utilisateur admin avec tous les droits
CREATE USER 'gsb_frais'@'localhost' IDENTIFIED BY 'P@ssw0rdGSB';
GRANT ALL PRIVILEGES ON gsb_frais.* TO 'gsb_frais'@'localhost';
```

Étape 3 : Création de l'utilisateur admin avec tous les droits

```
CREATE USER 'gsb_frais'@'localhost' IDENTIFIED BY 'P@ssw0rdGSB';
GRANT ALL PRIVILEGES ON gsb_frais.* TO 'gsb_frais'@'localhost';
```

Étape 4 : Création de l'utilisateur applicatif (consultation + mise à jour)

```
CREATE USER 'userGsb'@'localhost' IDENTIFIED BY 'secret';
GRANT SELECT, UPDATE ON gsb_frais.* TO 'userGsb'@'localhost';
```

Étape 5 : Appliquer les changements de droits

```
FLUSH PRIVILEGES;
```

Étape 6 : Importation des fichiers SQL

```
exit
mysql -u gsb_frais -p gsb_frais < gsb_frais_structure.sql
mysql -u gsb_frais -p gsb_frais < gsb_frais_insert_tables_statiques.sql
```

L
Y
C
MARGUERITE
JAUZELON

Mode opératoire Apache

ÉTAPE 1 : Mise à jour du système

Avant toute installation, il est important de mettre à jour la base de données des paquets afin d'avoir accès aux dernières versions stables des logiciels disponibles pour Debian.

```
apt update && apt upgrade -y
```

ÉTAPE 2 : Installation du serveur web Apache

Cette commande installe les services Apache2, php et Mysql les plus utilisés pour les systèmes Linux. Il s'agit uniquement de versions stables et précompilées, livrées avec les modules de base nécessaires.

```
apt install apache2 php libapache2-mod-php mysql-server php-mysql
```

module php :

```
apt install php-curl php-gd php-intl php-json php-mbstring php-xml php-zip
```

ÉTAPE 3 : Vérification

On vérifie que le service est bien actif et fonctionne correctement après l'installation.

```
systemctl status apache2
```

Si Apache est inactif, on peut le démarrer manuellement :

```
systemctl start apache2
systemctl enable apache2
```

ÉTAPE 4 : Vérification du fonctionnement via le navigateur

Depuis un autre poste ou le navigateur du serveur, accéder à l'adresse suivante :

http://172.18.155.82

Si Apache est bien installé, vous verrez une page d'accueil "Apache2 Debian Default Page".

ÉTAPE 5 : Configuration du répertoire web

Par défaut, Apache utilise le répertoire suivant pour héberger les sites web :

/var/www/html

Mais nous allons utiliser le répertoire suivant :

/var/www/appliGSB

```
mkdir /var/www/appliGSB
usermod -aG www-data sio
chown -R www-data:www-data /var/www/appliGSB
chmod -R 755 /var/www/appliGSB
```

ÉTAPE 6 : Gestion des sites virtuels

Pour héberger le sites ou séparer la configuration, ont vas créer un VirtualHost :

```
nano /etc/apache2/sites-available/appliGSB.conf
```

```
<VirtualHost *:80>

    Redirect permanent / https://gestionfraisintralabfls.gsb.coop/
    ServerAdmin webmaster@localhost
    ServerName gestionfraisintralabfls.gsb.coop
    ServerAlias www.gestionfraisintralabfls.gsb.coop
    DocumentRoot /var/www/appliGSB/appliFrais_avecMysqli
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

On vas ensuite transféré les fichiers web sur le serveurs via la commande scp

```
scp c:\Users\lsf\Downloads\appliFrais_avecMysqli sio@172.18.155.82:/var/www/appliGSB
scp c:\Users\lsf\Downloads\bddAppliFrais.zip sio@172.18.155.82:/var/www/appliGSB
```

Activation du site et rechargement d'Apache

```
a2ensite appliGSB.conf
a2dissite 000-default.conf
systemctl reload apache2
```

ÉTAPE 7 : [HTTPS] : Mise à jour des paquets

```
apt update && apt upgrade -y
```

ÉTAPE 8 : Activer les modules nécessaires pour HTTPS

ssl : permet de gérer les connexions sécurisées
headers : utile pour ajouter des en-têtes HTTP strictes (bonnes pratiques de sécurité)

```
a2enmod ssl
a2enmod headers
```

ÉTAPE 9 : Créer un certificat SSL auto-signé pour le serveur

Le certificat sera valide 1 an (365 jours)

```
mkdir -p /etc/ssl/appliGSB/
openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/appliGSB/intralablsf.key \
-out /etc/ssl/appliGSB/intralablsf.crt
```

ÉTAPE 10 : Créer un hôte virtuel HTTPS

On définit : un VirtualHost sur le port 443 avec le certificat SSL

```
nano /etc/apache2/sites-available/appliGSB_ssl.conf
```

```
<VirtualHost *:443>
    ServerName gestionfraisintralabfls.gsb.coop
    DocumentRoot /var/www/appliGSB/appliFrais_avecMysqli

    SSLEngine on
    SSLCertificateFile /etc/ssl/appliGSB/intralablsf.crt
```

```
    SSLCertificateKeyFile /etc/ssl/appliGSB/intralablsf.key

    <Directory /var/www/appliGSB/appliFrais_avecMysqli>
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

ÉTAPE 11 : Activer le site HTTPS et désactiver éventuellement le HTTP par défaut

```
a2ensite appliGSB_ssl.conf
a2dissite appliGSB.conf
```

ÉTAPE 12 : Redémarrer Apache pour appliquer la configuration

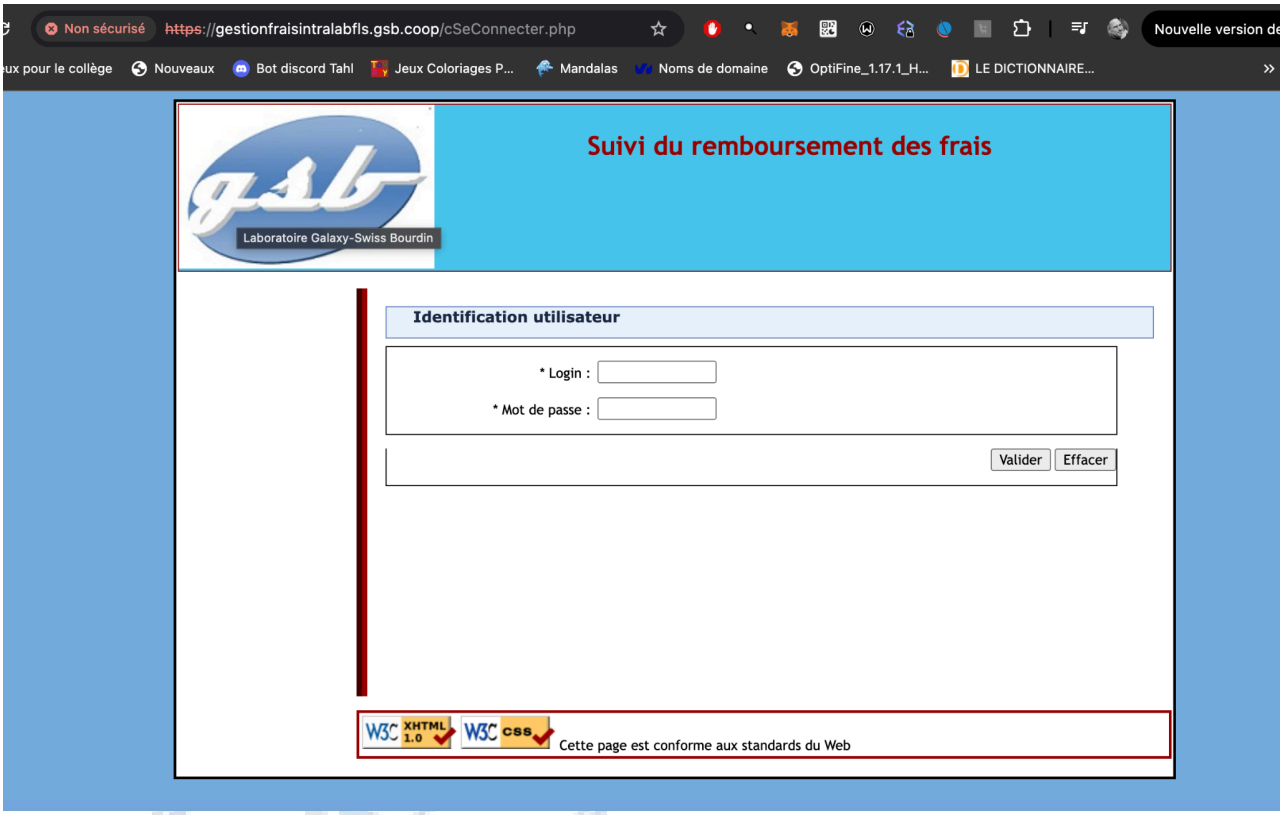
```
systemctl reload apache2
```

ÉTAPE 13 : Vérifier le bon fonctionnement

Ouvre ton navigateur et va à l'adresse :

https://172.18.155.82

Un avertissement de sécurité peut apparaître à cause du certificat auto-signé (normal)



ÉTAPE 13 : Apache headers

à rajouté dans le fichier -> /etc/apache2/sites-available/appliGSB_ssl.conf

```
 Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains; preload"
    Header always set X-Frame-Options "SAMEORIGIN"
    Header always set X-Content-Type-Options "nosniff"
    Header always set Referrer-Policy "no-referrer-when-downgrade"
    Header always set Permissions-Policy "geolocation=(), microphone=(), camera=()"
    Header always set X-XSS-Protection "1; mode=block"
    Header always set Content-Security-Policy "default-src 'self'; script-src 'self'; style-src 'self'; img-src 'self' data:; font-src 'self'; object-src 'none'; base-uri 'self'; form-action 'self'"
```

Ce que chaque en-tête fait :

- **Strict-Transport-Security** : Forcer HTTPS pour 2 ans + sous-domaines (utilisé pour HSTS preload).
- **X-Frame-Options** : Empêche l'affichage du site dans une iframe (clickjacking).
- **X-Content-Type-Options** : Empêche la détection automatique du type MIME.
- **Referrer-Policy** : Contrôle les données du référent envoyées à d'autres sites.
- **Permissions-Policy** : Bloque l'accès aux API sensibles (géolocalisation, micro, caméra).
- **X-XSS-Protection** : Active le filtre XSS côté navigateur (utile avec anciens navigateurs).
- **Content-Security-Policy (CSP)** : Politique très stricte qui bloque les ressources externes non autorisées (scripts, styles, images, etc.).

Mode opératoire HA (Corosync & pacemaker)

<u>Étape 1</u> : Installation et configuration de Corosync et Pacemaker

apt update
apt install corosync pacemaker crmsh

Vérification de la ressource : service corosync status

```
root@intralabFLS:/home/sio# service corosync status
● corosync.service - Corosync Cluster Engine
     Loaded: loaded (/lib/systemd/system/corosync.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-04-06 14:53:55 +04; 1 months 2 days ago
       Docs: man:corosync
             man:corosync.conf
             man:corosync_overview
   Main PID: 432 (corosync)
      Tasks: 9 (limit: 1115)
     Memory: 159.7M
        CPU: 6h 1min 58.883s
     CGroup: /system.slice/corosync.service
             └─432 /usr/sbin/corosync -f
```

Création d'un fichier '' authkey '' avec la commande " corosync-keygen " ce fichier doit être présent sur tous les nœuds du cluster.

Il faut désormais cloner la VM " Maître " pour pouvoir créer le cluster.

Définir un mot de passe pour l'utilisateur hacluster

passwd hacluster

Éditer le fichier corosync.conf :

nano /etc/corosync/corosync.conf sur le Master et slave

On vas venir modifier les variable suivantes :

bindnetaddr: 172.18.155.0
mcastaddr: 239.255.1.8

```
nodelist {
  node {
    name: wan-master
    nodeid: 1
    ring0_addr: 172.18.155.82
  }
  node {
    name: wan-slave
    nodeid: 2
    ring0_addr: 172.18.155.83
  }
}
```

On va démarrer tout les services et faire un crm status pour vérifier la configuration :

systemctl restart corosync
systemctl enable corosync
crm status

```
root@intralabFLS:/home/sio# crm status
Cluster Summary:
  * Stack: corosync
  * Current DC: wan-master (version 2.0.5-ba59be7122) - partition with quorum
  * Last updated: Fri May  9 14:34:51 2025
  * Last change:  Sun Apr  6 14:47:40 2025 by root via crm_attribute on wan-master
  * 2 nodes configured
  * 1 resource instance configured

Node List:
  * Online: [ wan-master wan-slave ]

Full List of Resources:
  * IPFailover  (ocf::heartbeat:IPaddr2):        Started wan-master

root@intralabFLS:/home/sio#
```

Étape 2 : Désactivation de stonith

C'est en fait un mécanisme pour éteindre complètement le serveur qui vient de flancher en éteignant son onduleur. C'est un procédé surtout utilisé avec des disques partagés car il serait dangereux que l'ordinateur qui est supposé être hors d'état vienne écrire sur le disque partagé et corrompre/altérer les données.

crm configure property stonith-enabled=false

La vérification (root@intralabXX:~# crm_verify -L -V) ne renvoie plus d'erreur.

Étape 3 : configuration de l'ip flottante (IPFaillover)

root@intralabXX:~# crm configure primitive IPFailover ocf:heartbeat:IPaddr2 params ip=172.18.155.84 cidr_netmask=21 nic=ens192 iflabel=VIP

· **primitive** : argument pour ajouter une primitive regroupant plusieurs valeurs indiquant au Cluster quels scripts utiliser pour la ressource, où le trouver et à quel standard il correspond.

· **ocf** : classe de la ressource (ça pourrait donc aussi être lsb) · heartbeat : fournisseur de la ressource

· **IPaddr2** : ressource gérant les adresses IPv4 virtuelles ==> le script appelé

· **params** : déclaration des paramètres nécessaires à la ressource · IPFailover : le nom de la ressource (il est évidemment libre... mais doit être suffisamment « parlant »),

· **IPaddr2** : le script appelé · params : suivent les différents paramètres à appliquer

· **ip**=17218.155.84 : nom et valeurs du paramètre « ip »

· **cidr_netmask=21** : masque de sous-réseau en notation CIDR

· **nic=ens192** : carte réseau sur laquelle est appliquée l'adresse IP virtuelle

· **iflabel=VIP** : permet de donner un label (étiquette) à la carte réseau virtuelle. Sans ce label, la VIP n'est pas visible avec la commande ifconfig mais seulement avec la commande ip addr show

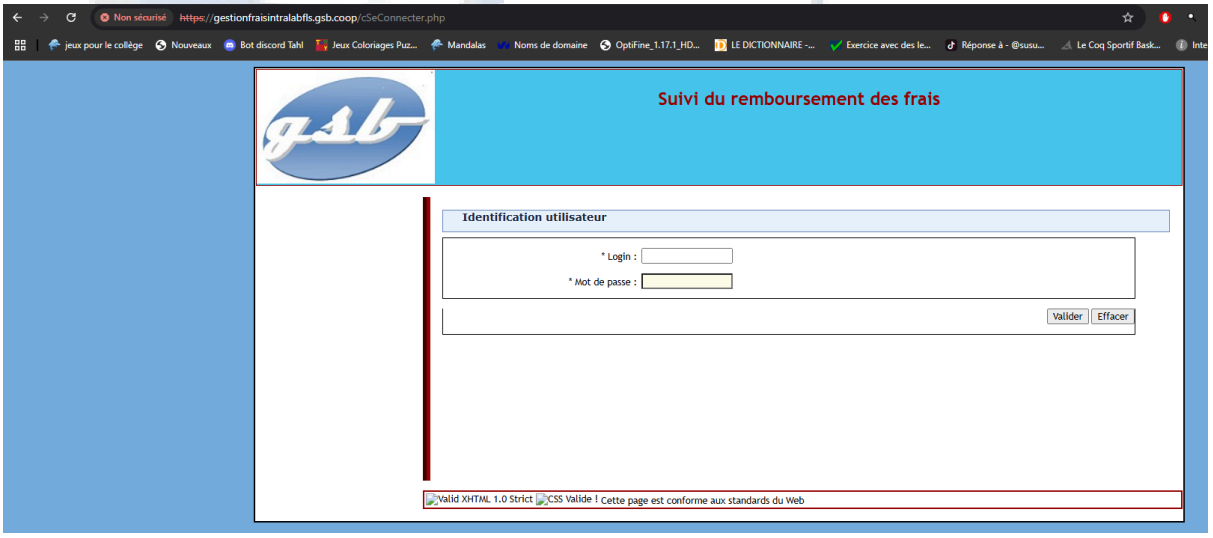
Vérification : crmstatus

```
root@intralabFLS:/home/sio# crm status
Cluster Summary:
  * Stack: corosync
  * Current DC: wan-master (version 2.0.5-ba59be7122) - partition with quorum
  * Last updated: Fri May  9 14:45:29 2025
  * Last change:  Sun Apr  6 14:47:40 2025 by root via crm_attribute on wan-master
  * 2 nodes configured
  * 1 resource instance configured

Node List:
  * Online: [ wan-master wan-slave ]

Full List of Resources:
  * IPFailover  (ocf::heartbeat:IPaddr2):        Started wan-master
```

cmd windows : nslookup gestionfraisintralabfls.gsb.coop

```
C:\Users\Florian>nslookup gestionfraisintralabfls.gsb.coop
Serveur :   ADW2019S.sio.lan
Address:  172.18.159.250

Réponse ne faisant pas autorité :
Nom :    gestionfraisintralabfls.gsb.coop
Address:  172.18.155.84
```

site web gestionfraisintralabfls.gsb.coop :



Étape 4 : Réplication de base de donnée

Sur le serveur MASTER :

Configurer le fichier /etc/mysql/mariadb.conf.d/50-server.cnf

- commenter la ligne bind-address = 127.0.0.1
- Avoir différant server-id sur le master - slave
- Décommenter la ligne log_bin = /var/log/mysql/mysql-bin.log
- Décommenter la ligne #max_binlog_size = 100M
- Rajouter la ligne binlog_do_db = gsb_frais

Bloquer l'écriture via la commande sous mariadb : FLUSH TABLES WITH READ LOCK;

Vérifier que la base de donnée est bien répliqué : show master status;

il faut noter les champs " file " et " position " -> mysql-bin.000001 ; 3921

Redémarer mariadb

—------

Sur le serveur SLAVE :

Configurer le fichier /etc/mysql/mariadb.conf.d/50-server.cnf

- Avoir différant server-id sur le master - slave
- Décommenter la ligne #max_binlog_size = 100M
- Rajouter la ligne master-retry-count = 20
- Rajouter la ligne replicate-do-db = gsb_frais

Sous mariadb :

mysql -u root -p

stop slave;

change master to master_host='172.18.155.82', master_user='gsb_frais', master_password='P@ssw0rdGSB', master_log_file='mysql-bin.000001', master_log_pos=3921;

start slave ;

Redémarer mariadb

En dernier sur le serveur MASTER : il faut débloquer les tables sur mysql : UNLOCK TABLES;

Étape 5 : Réplication de base de donnée :

Configuration du fichier /etc/mysql/mariadb.conf.d/50-server.cnf - **MASTER**

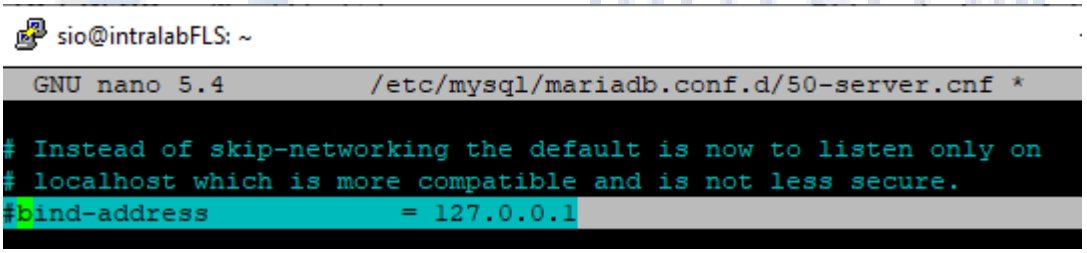
- commenter la ligne bind-address

- server-id 1

- Décommenter la ligne #max_binlog_size = 100M

- Rajouter la ligne master-retry-count = 20

- Rajouter la ligne replicate-do-db = gsb_frais



```
sio@intralabFLS: ~
  GNU nano 5.4          /etc/mysql/mariadb.conf.d/50-server.cn
#long_query_time         = 10
#log_slow_verbosity      = query_plan,explain
#log-queries-not-using-indexes
#min_examined_row_limit = 1000

# The following can be used as easy to replay backup logs or
# note: if you are setting up a replication slave, see README
#       other settings you may need to change.
server-id               = 1
log_bin                 = /var/log/mysql/mysql-bin.log
expire_logs_days         = 10
max_binlog_size         = 100M
binlog_do_db = gsb_frais
#
```

Après avoir effectué une résolution de problème, il faut impérativement relancer mysql si non les configurations ne seront pas prise en compte et les commandes suivantes ne fonctionnent pas.

On se met sur la base de donnée “ gsb_frais “

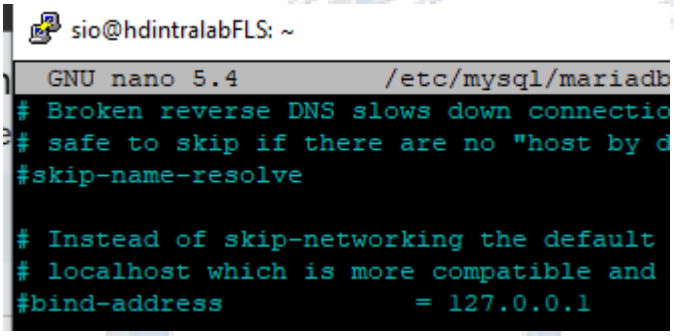
```
Database changed
MariaDB [gsb_frais]> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0,004 sec)

MariaDB [gsb_frais]>
```

On tape “ FLUSH TABLES WITH READ LOCK; “ pour bloquer l’écriture le temps de la configuration de la réplication. On note les champs “ file “ et “ position “ pour plus tard

```
MariaDB [(none)]> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> show master status;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |      328 | gsb_frais    |                  |
+------------------+----------+--------------+------------------+
1 row in set (0,000 sec)
```

Configurer le fichier /etc/mysql/mariadb.conf.d/50-server.cnf - **SLAVE**

- server-id 2
- Décommenter la ligne #max_binlog_size = 100M
- Rajouter la ligne master-retry-count = 20
- Rajouter la ligne replicate-do-db = gsb_frais

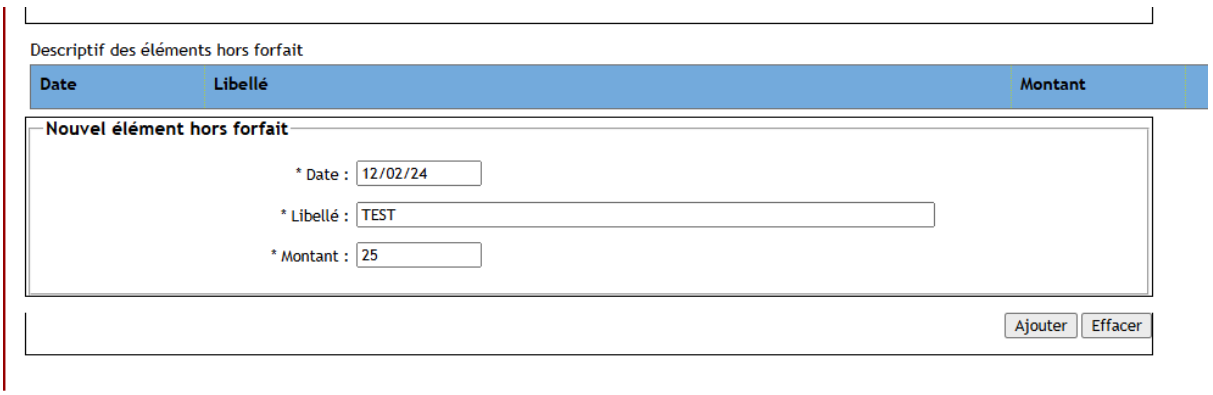


Étape 6 : Testing

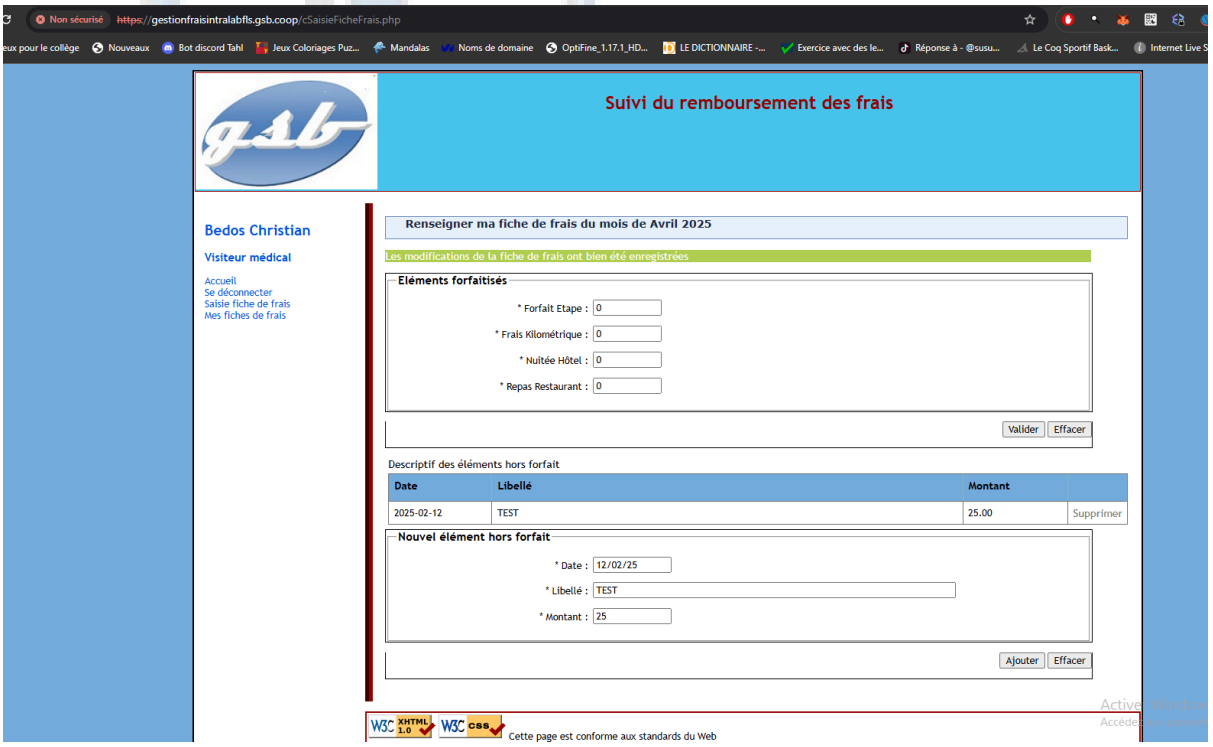
Pour effectuer tout les test, j'ai utilisé le compte de Bedos Christian :

J'ai stoppé le serveur Master car il était en Current DC

Puis dans la section “ saisie fiche de frais “, j’ai pu en saisir une :



Après avoir rallumer le Master, l'enregistrement de la fiche de frais est bien visible sur le Master. La réplication est bien fonctionnel dans les deux sens :



Mode opératoire Centreon

Étape 1 – Mise à jour du système

```
apt update && apt upgrade -y
```

Étape 2 – Installer les dépendances principales

```
apt install -y apache2 mariadb-server php php-mysql php-xml php-gd php-curl php-intl
php-mbstring php-zip php-bcmath php-soap libapache2-mod-php wget curl unzip gnupg2
```

Étape 3 – Configuration de MariaDB

```
mysql_secure_installation

mysql -u root -p

CREATE DATABASE centreon CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE USER 'centreon'@'localhost' IDENTIFIED BY 'centreon';
GRANT ALL PRIVILEGES ON centreon.* TO 'centreon'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Étape 4 – Configuration d'Apache

```
a2enmod rewrite ssl
systemctl restart apache2
```

Étape 5 – Installation de Centreon

```
cd /tmp
wget https://download.centreon.com/standard/21.10/centreon-21.10.13.tar.gz
tar -xzf centreon-21.10.13.tar.gz
mv centreon-21.10.13 /usr/local/centreon

useradd -m -s /bin/bash centreon
chown -R centreon:centreon /usr/local/centreon
```

Étape 6 – Configuration PHP pour Centreon

```
nano /etc/php/7.4/apache2/php.ini

memory_limit = 256M
upload_max_filesize = 100M
post_max_size = 100M
max_execution_time = 300
date.timezone = Europe/Paris
```

Étape 7 – Accès web

```
nano /etc/apache2/sites-available/centreon.conf

<VirtualHost *:80>
    ServerAdmin admin@centreon.local
    DocumentRoot /usr/local/centreon/www
    ServerName centreon.local

    <Directory /usr/local/centreon/www>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/centreon_error.log
    CustomLog ${APACHE_LOG_DIR}/centreon_access.log combined
</VirtualHost>

a2ensite centreon
systemctl reload apache2
```

Étape 8 – Lancer l'installation via navigateur

Accédez à `http://172.18.155.86` ou `http://centreon.local`

Suivez l'installation graphique

- Base de données : `centreon / centreon`
- Dossier : `/usr/local/centreon`

Étape 9 – Post-installation

Démarrer les services :

```
/usr/local/centreon/bin/cbd
/usr/local/centreon/bin/centcore
/usr/local/centreon/bin/centengine
```

Ajouter à /etc/rc.local (si existant) ou créer un service systemd

Installez **NRPE** pour que Centreon puisse interroger les services à distance :

```
apt install -y nagios-nrpe-server nagios-plugins
```

Ajoutez dans /etc/nagios/nrpe.cfg

```
allowed_hosts=127.0.0.1,172.18.155.86
```

Ajoutez les commandes :

```
command[check_apache]=/usr/lib/nagios/plugins/check_http -p 80
command[check_mariadb]=/usr/lib/nagios/plugins/check_mysql -u root
```

Puis redémarrez le service :

```
systemctl restart nagios-nrpe-server
```

il faut maintenant répétez ces étapes sur 172.18.155.82 et .83

Configuration des hôtes dans Centreon

Fichier de configuration Centreon (Nagios) – `/etc/centreon-engine/hosts.cfg`

```
define host {
    use                 generic-host
    host_name               master
    alias               Serveur MASTER
    address               172.18.155.82
}

define host {
    use                 generic-host
    host_name               slave
    alias               Serveur SLAVE
    address               172.18.155.83
}
```

```
define host {
    use                 generic-host
    host_name               ip_virtual
    alias               IP Failover
    address               172.18.155.84
}
```

Configuration des services – /etc/centreon-engine/services.cfg

```
# Vérification Apache sur master

define service {
    use                 generic-service
    host_name               master
    service_description     Apache
    check_command           check_nrpe!check_apache
}

# Vérification MariaDB sur master
define service {
    use                 generic-service
    host_name               master
    service_description     MariaDB
    check_command           check_nrpe!check_mariadb
}

# Vérification Apache sur slave
define service {
    use                 generic-service
    host_name               slave
    service_description     Apache
    check_command           check_nrpe!check_apache
}

# Vérification MariaDB sur slave
define service {
    use                 generic-service
    host_name               slave
    service_description     MariaDB
    check_command           check_nrpe!check_mariadb
}

# Vérification de l'IP virtuelle en ping
define service {
    use                 generic-service
    host_name               ip_virtual
    service_description     Ping
```

```
    check_command           check_ping!100.0,20%!500.0,60%
}

systemctl restart centreon-engine
```

Mode opératoire Bacula

Ajoutez le dépôt officiel de Bacula :

Instalation sur Debian 12 :

apt install software-properties-common -y

add-apt-repository 'deb http://bacula.org/packages/debian/ focal main'

wget -O - http://www.bacula.org/downloads/Bacula-4096-Distribution-Verification-key.asc | apt-key add -

apt update

INSTALATION BACULA SERVER :

apt install bacula-server

Installation de Bacula sur Debian 12

Voici les étapes pour installer **Bacula** sur Debian 12 avec l'IP 172.18.154.142 :

Étape 1 : Mettre à jour le système

```
apt update && apt upgrade -y
```

Étape 2 : Installer le serveur Bacula

Ajoutez le dépôt officiel de Bacula :

```
apt install software-properties-common -y
add-apt-repository 'deb http://bacula.org/packages/debian/ focal main'
```

```
wget -O -
http://www.bacula.org/downloads/Bacula-4096-Distribution-Verificatio
n-key.asc | apt-key add -
apt update
```

Installez le serveur Bacula et ses outils :

```
apt install bacula-server bacula-client -y
```

Étape 3 : Configurer Bacula

Les fichiers principaux à configurer sont :

- **/etc/bacula/bacula-dir.conf** (Director)
- **/etc/bacula/bacula-sd.conf** (Storage Daemon)
- **/etc/bacula/bacula-fd.conf** (File Daemon)

Ouvrez le fichier **bacula-dir.conf** :

```
nano /etc/bacula/bacula-dir.conf
```

1. Configurez votre IP (172.18.154.142) pour les sections **Director** et **File Daemon**.

Modifiez **bacula-sd.conf** pour spécifier le répertoire de stockage et l'IP :
bash
Copier le code
```
nano /etc/bacula/bacula-sd.conf
```

2. Assurez-vous que l'adresse du Storage Daemon est bien configurée avec `172.18.154.142`.

Configurez **bacula-fd.conf** pour accepter les connexions du Director :
bash
Copier le code
```
nano /etc/bacula/bacula-fd.conf
```

3. Spécifiez l'adresse du Director comme étant `172.18.154.142`.

Étape 4 : Créer une base de données pour Bacula

Installez MariaDB :

bash
Copier le code
```
apt install mariadb-server -y
```

Créez la base de données Bacula :

bash
Copier le code

```
mysql -u root -p
CREATE DATABASE bacula;
CREATE USER 'bacula'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON bacula.* TO 'bacula'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Initialisez la base de données avec le script fourni par Bacula :

SI les fichier ne sont pas là il faut utiliser cette commande

```
/usr/lib/bacula/create_mysql_database
/usr/lib/bacula/make_mysql_tables
/usr/lib/bacula/grant_mysql_privileges
```

sI les fichier ne sont pas là il faut utilisé cette commande :

```
find / -type f -name "create_mysql_database*" 2>/dev/null
```

Étape 5 : Redémarrer les services Bacula

```
systemctl restart bacula-director
systemctl restart bacula-sd
systemctl restart bacula-fd
```

Étape 6 : Vérifier l'installation

Utilisez l'outil **bconsole** pour vérifier la connexion :

```
bconsole
```

L'interface CLI doit vous afficher un prompt vous permettant de gérer Bacula.

Étape 7 : Ajouter des tâches de sauvegarde

Ajoutez vos clients et tâches dans le fichier **bacula-dir.conf** sous la section `Job` et `FileSet`.

Préparer les scripts de dump MariaDB sur les clients

Sur `172.18.155.82` et `172.18.155.83`, crée ce script :

Fichier : `/opt/bacula/scripts/backup_mariadb.sh`

```
#!/bin/bash
DATE=$(date +\%F)
mkdir -p /var/backups/mariadb
mysqldump -u root --password='12-Soleil&' --all-databases >
/var/backups/mariadb/mariadb_$DATE.sql
```

chmod +x /opt/bacula/scripts/backup_mariadb.sh

Configuration dans `/etc/bacula/bacula-dir.conf` (sur le serveur Bacula)

```
Client {
  Name = master-fd
  Address = 172.18.155.82
  FDPort = 9102
  Catalog = MyCatalog
  Password = "masterpass"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}

Client {
  Name = slave-fd
  Address = 172.18.155.83
  FDPort = 9102
  Catalog = MyCatalog
  Password = "slavepass"
  File Retention = 30 days
  Job Retention = 6 months
  AutoPrune = yes
}
```

```
FileSet {
```

```
  Name = "MariaDB-Master"
  Include {
    Options {
      signature = MD5
    }
    File = /var/backups/mariadb
  }
}

FileSet {
  Name = "MariaDB-Slave"
  Include {
    Options {
      signature = MD5
    }
    File = /var/backups/mariadb
  }
}

JobDefs {
  Name = "DefaultJob"
  Type = Backup
  Level = Incremental
  Schedule = "WeeklySat"
  Storage = File
  Messages = Standard
  Pool = Default
  Priority = 10
}

Job {
  Name = "Backup-MariaDB-Master"
  Client = master-fd
  JobDefs = "DefaultJob"
  FileSet = "MariaDB-Master"
  RunBeforeJob = "/opt/bacula/scripts/backup_mariadb.sh"
}

Job {
  Name = "Backup-MariaDB-Slave"
  Client = slave-fd
  JobDefs = "DefaultJob"
  FileSet = "MariaDB-Slave"
  RunBeforeJob = "/opt/bacula/scripts/backup_mariadb.sh"
}


Schedule {
```

```
  Name = "WeeklySat"
  Run = Full 1st sat at 02:00
}
```

—-------

```
systemctl restart bacula-director
systemctl restart bacula-sd
systemctl restart bacula-fd
```

—---------

```
bconsole
*run
```

Annexe :

Schéma infrastructure GSB :