

Compte rendu d'activité

TP : Mise en place d'un environnement de développement
sous machine virtuelle

03/2022

Contexte

Travaillant dans une entreprise de services numériques (ESN) qui accompagne ses clients dans la mise en œuvre de solutions numériques et mobiles depuis leur conception jusqu'à leur intégration dans le système d'information du client, il est nécessaire de mettre en place le déploiement d'une application cliente, à sa validation technique et à sa mise en production. Vous assistez les utilisateurs dans la prise en main des nouvelles solutions logicielles mises en place.

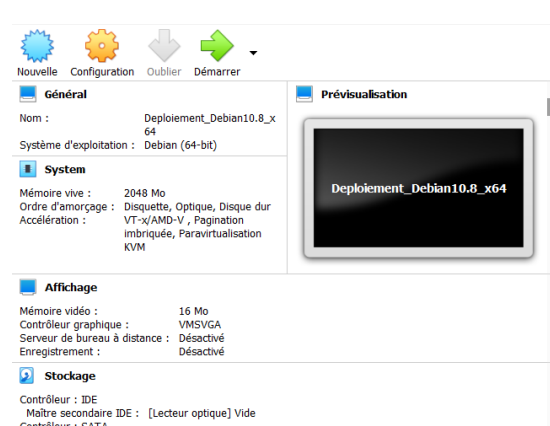
Expressions des besoins

La solution applicative a été développée et son code source est disponible. Il est demandé de la tester en environnement de préproduction en utilisant les mêmes outils que ceux prévus en production. Le code source de l'application est accessible à ce [lien](#). Une machine virtuelle Debian a été fournie, qu'il convient de configurer.

Realisation

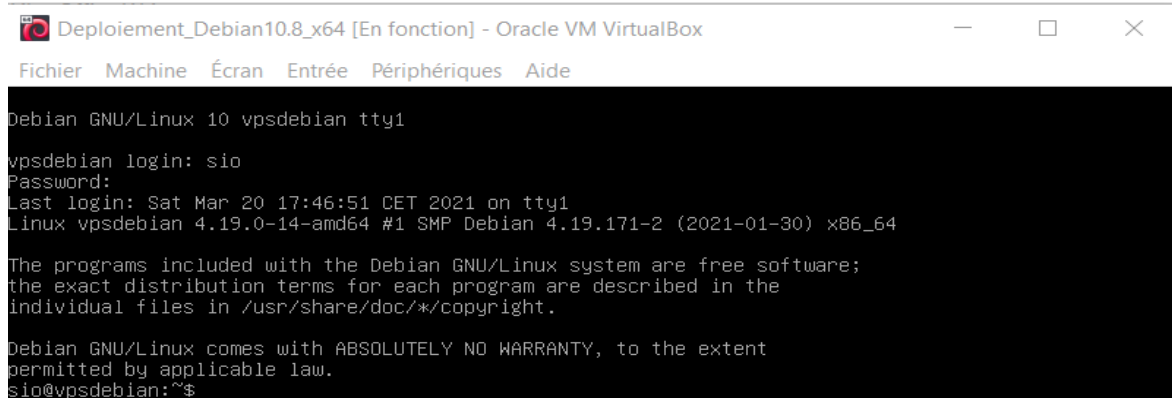
Configurations serveur

Après importation de la machine virtuelle Debian fournie, il suffit de la démarrer.



Après chargement, l'écran suivant, s'affiche demandant d'entrer un login puis mot de passe, ces derniers sont fournis dans la description de la VM. Ainsi, il convient de les rentrer dans la console.

Mise à jour système



```
Deploiement_Debian10.8_x64 [En fonction] - Oracle VM VirtualBox
Fichier Machine Écran Entrée Périphériques Aide

Debian GNU/Linux 10 vpsdebian tty1
vpsdebian login: sio
Password:
Last login: Sat Mar 20 17:46:51 CET 2021 on tty1
Linux vpsdebian 4.19.0-14-amd64 #1 SMP Debian 4.19.171-2 (2021-01-30) 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.
sio@vpsdebian:~$
```

Il est nécessaire de faire une mise à jour des paquets de la VM par la commande suivante :

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

où

- `sudo` : permet de demander une élévation de privilège en tant que super-utilisateur .
- `apt` : gestionnaire des paquets sous Debian (acronyme de « Advanced Packaging Tool »).
- `-y` : permet de ne pas avoir une demande de confirmation pour la réalisation de cette opération.

Faire ensuite la mise à jour des paquets, à partir du catalogue que l'on vient de mettre à jour avec `upgrade`

Installer le serveur

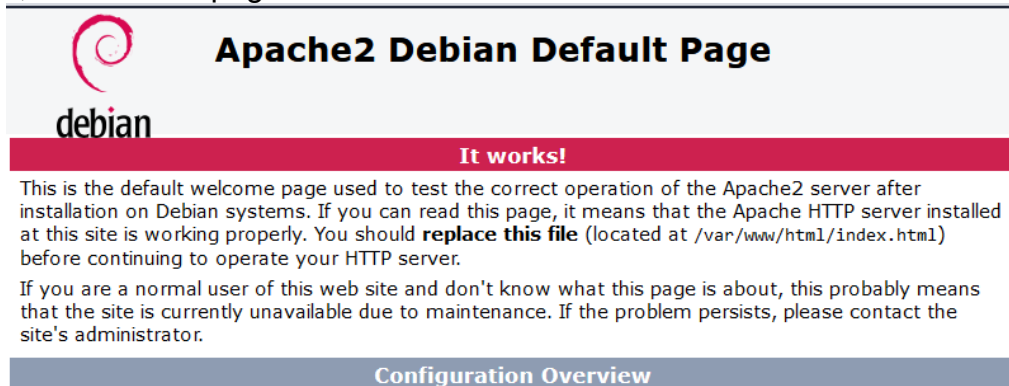
Le serveur installé sera celui d'Apache HTTP Server qui se fait par la commande suivante :

```
sudo apt -y install apache2
```

Afin de vérifier que l'opération ait produit le résultat attendu, il faut vérifier que le port 80 (http) est bien dans la liste des ports ouverts sur le serveur

```
LISTEN 0 128 0.0.0.0:80 0.0.0.0:80 http 0.0.0.0:80 0.0.0.0:80
```

Ainsi, le port est ouvert et lorsqu'on se rend à l'adresse ip du serveur dans un navigateur, on obtient la page suivante :



l'adresse IP est récupérée avec la commande : **ip a**

Installer php

La commande nécessaire à cette installation est :

```
sudo apt -y install php
```

Pour vérifier son installation, il est possible d'effectuer la commande suivante qui permet d'afficher dans la console la version de php installée.

```
php -v
```

```
sio@LAMP-deb10-buster:~$ php -v
PHP 7.3.19-1~deb10u1 (cli) (built: Jul  5 2020 06:46:45) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.19, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.19-1~deb10u1, Copyright (c) 1999-2018, by Zend Technologies
sio@LAMP-deb10-buster:~$ _
```

Puis, en effectuant la commande suivante, il est possible d'écrire le résultat de la commande phpinfo, afin de pouvoir l'ouvrir dans l'url du serveur Apache. Cela permet ainsi de savoir si l'interpréteur Php fonctionne correctement.

```
Sudo bash -c 'echo « <?php phpinfo() ; » > var/www/html/test.php'
```

Où les instructions :

- -c » : permet de regrouper toute la commande entre simples quotes comme une seule commande à passer à sudo
- phpinfo() : fonction php qui produit l'affichage de nombreuses informations sur sa configuration de PHP.
- > : permet de rediriger le résultat de la commande echo à l'intérieur du fichier mentionné à droite.
- /var/www/html : le répertoire d'hébergement par défaut du serveur Apache.


PHP Version 7.3.19-1~deb10u1 	
System	Linux LAMP-deb10-buster 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64
Build Date	Jul 5 2020 06:46:45
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqld.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2

Figure 1: Affichage du fichier test.php dans le navigateur

Installer le SGBD MariaDb

La commande nécessaire à cette installation est

```
sudo apt -y install mariadb-server
```

Pour s'assurer de sa bonne installation, il est possible de tester l'accès serveur avec les commandes suivantes, la première permet son ouverture :

```
sudo mariadb -u root
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.3.34-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Il est aussi possible de tester si l'on a bien accès aux bases de données du SGBD en demandant l'affichage de la liste des bases de données, par la commande suivante :

```
SHOW DATABASES ;
```

Ainsi, il est possible d'observer que l'ensemble des bases de données disponibles sont bien présentes, dont chocolatein qui est celle de l'application cible.

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| chocolatein |
| dvwa      |
| information_schema |
| mysql     |
| nolark    |
| performance_schema |
+-----+
5 rows in set (0.014 sec)
```

Le compte administrateur « root » ne fonctionne qu'en local sur la machine. Il est toutefois possible de créer un nouveau compte qui permettra de se connecter au serveur depuis un script PHP interprété par le serveur web.

Création d'un administrateur MariaDB

Pour ce faire, commencez par vous connecter à la console MariaDB comme on l'a fait ci-avant. Ensuite, saisissez les instructions SQL suivantes :

```
CREATE USER 'adminmariadb'@'localhost' IDENTIFIED BY 'motdepasse';
GRANT ALL PRIVILEGES ON *.* TO 'adminmariadb'@'localhost' WITH GRANT
OPTION;
FLUSH PRIVILEGES;
```

Installation pilotes mysql

Il est possible de récupérer les pilotes d'accès à MariaDb avec la commande suivante :

```
sudo apt -y install php-mysql
```

Puis, il est nécessaire de rafraîchir la configuration

```
sudo service apache2 reload
```

Installation de phpMyAdmin

PhpMyAdmin est une application web en PHP permettant d'administrer MySQL (PHP MySQL Administrator) et MariaDB. Il est possible de le télécharger par la commande suivante :

```
wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

Où, wget est un utilitaire permettant de récupérer une ressource web (Web GET) à partir de son URL.

Puis, il faut décompresser son archive

```
tar -xzf phpMyAdmin-latest-all-languages.tar.gz --one-top-level=phpmyadmin --strip-components=1
```

Où tar : est un programme d'achivage (Tape ARhiver) qui permet de compresser/décompresser des contenus.

Puis, il faut installer les extensions nécessaires à son fonctionnement :

```
sudo apt -y install php-json php-mbstring php-zip php-gd php-xml php-curl
```

Une fois ces extensions installées, ne pas oublier de recharger la configuration du serveur web avec l'instruction suivante :

```
sudo service apache2 reload
```

Installation d'OpenSSH Serveur

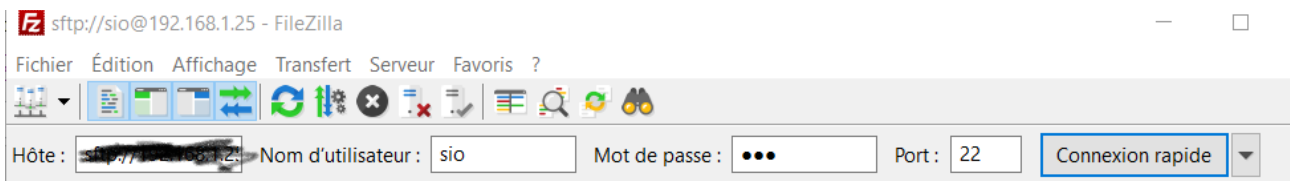
OpenSSH-Server permet de se connecter à un serveur de façon sécurisée en utilisant le protocole SSH en chiffrant la communication. Ce protocole permet également de transférer des fichiers. Pour l'installer, il suffit d'utiliser l'instruction suivante :

```
sudo apt -y install openssh-server
```

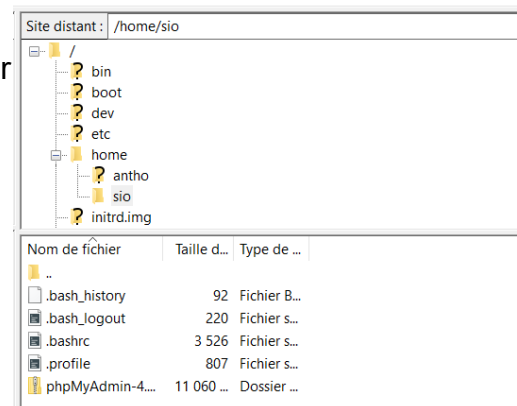
Connexion depuis le client SSH FileZilla Client

Il est nécessaire de télécharger ce logiciel depuis sa plateforme à ce [lien](#). Puis de suivre la procédure standard d'installation. A son ouverture, il suffit de se connecter avec les

informations de connexion au serveur dans la barre de connexion rapide, où l'adresse ip est récupérée depuis la console avec la commande **ip a**.



Une fois la connexion établie, il est possible de trouver le système de fichier local dans l'onglet de droite

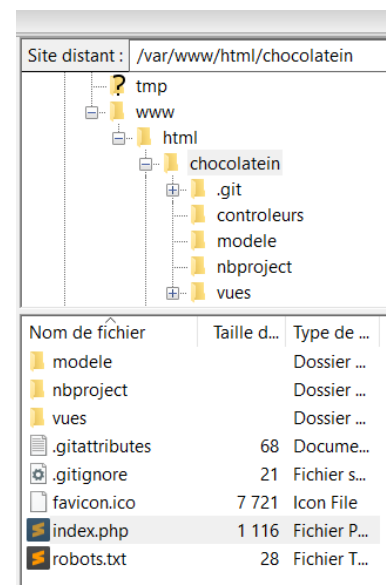


Configuration application

Dans FileZilla, il convient d'uploader le code source de l'application dans le dossier **/var/www/html**

Pour se rendre sur la page du site hébergé par le serveur adresse est construite ainsi :

http://adresseip/chocolatein/index.php



Il faut encore néanmoins reconstruire la base de données, grâce à phpMyAdmin préalablement téléchargé. Il convient alors de se rendre à l'adresse **http://adresseip:9000** où l'on arrive sur la fenêtre de connexion à **phpmyadmin**. Il suffit de se connecter avec les identifiants de connexion au serveur puis d'importer le script de récupération de la base de données.

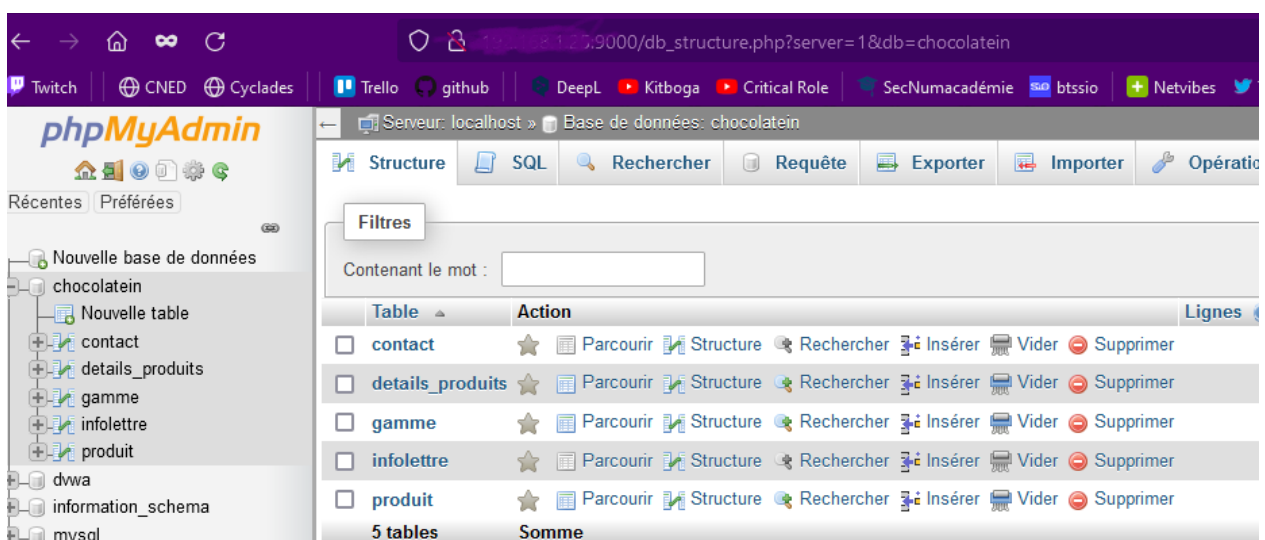
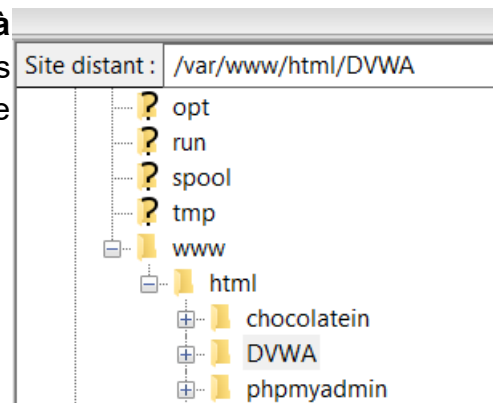


Figure 2: Affichage des des tables de la base de données chocolatein depuis le serveur



Figure 3: Affichage du site depuis l'adresse serveur avec ses données chargées

Liste des compétences couvertes

Gérer le patrimoine informatique :

- Recenser et identifier les ressources numériques
- Vérifier les conditions de la continuité d'un service informatique

Mettre à disposition des utilisateurs un service informatique :

- Déployer un service
- Accompagner les utilisateurs dans la mise en place d'un service

Compte rendu d'activité

TP : Utilisation de Github et aide utilisateur

03/2022

Contexte

Travaillant dans une entreprise de services numériques (ESN) qui accompagne ses clients dans la mise en œuvre de solutions numériques et mobiles depuis leur conception jusqu'à leur intégration dans le système d'information du client, il est nécessaire de mettre en place le déploiement d'une application cliente, à sa validation technique et à sa mise en production. Vous assistez les utilisateurs dans la prise en main des nouvelles solutions logicielles mises en place à partir du serveur créé sous machine virtuelle.

Expressions des besoins

- Il est souhaité que les mises à jour doivent se faire automatiquement avec Git.
- La nouvelle version permet de contrôler qu'un email saisie dans la partie abonnement ne comporte pas d'erreur dans son format. Il faut écrire les scénarios de tests permettant de tester cette nouvelle fonctionnalité.
- Mettre à jour le code actuellement sur le serveur à partir de la dernière version présente dans la « branche » de développement sur GitHub nommée « dev ». Pour ce faire, il faut donc remplacer les fichiers actuels, par ceux présents dans la dernière version de la branche « dev ».
- Après avoir Fork le projet, tester la nouvelle fonctionnalité de la branche dev de recherche accessible sur toutes les pages. Créer des issues depuis la plateforme github en cas d'anomalie détecté.
- Enfin, une vidéo de présentation de la nouvelle fonctionnalité sera mise en place avec le logiciel OBS pour une documentation utilisateur. Visible à ce [lien](#)

Informations techniques

Le code source de l'application est accessible à ce [lien](#).

Realisation

Récupération du code source avec git

Le code source de l'application étant accessible à ce [lien](#), un dossier de dépôt a été créé en local. Depuis l'invite de commande de git bash, ce dossier a été converti en dépôt par la commande : **git init**

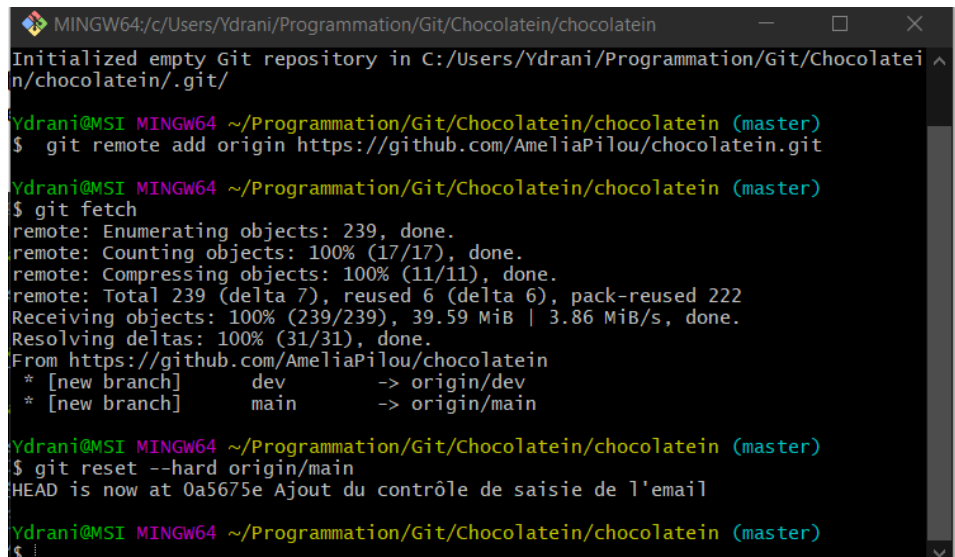
Ensuite, le code source a été récupéré par la commande de connexion **remote add**.

Puis, un téléchargement de la dernière version a été effectué avec : **git fetch**

Enfin, les fichiers de cette dernière version ont remplacé les anciens avec la commande :

git reset --hard origin/main

où **git reset** permet de forcer la mise à jour du dépôt local à partir de n'importe quelle branche d'un dépôt.



```
MINGW64: c:/Users/Ydrani/Programmation/Git/Chocolatein/chocolatein
Initialized empty Git repository in C:/Users/Ydrani/Programmation/Git/Chocolatein/chocolatein/.git/

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git remote add origin https://github.com/AmeliaPilou/chocolatein.git

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git fetch
remote: Enumerating objects: 239, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 239 (delta 7), reused 6 (delta 6), pack-reused 222
Receiving objects: 100% (239/239), 39.59 MiB | 3.86 MiB/s, done.
Resolving deltas: 100% (31/31), done.
From https://github.com/AmeliaPilou/chocolatein
 * [new branch]      dev       -> origin/dev
 * [new branch]      main      -> origin/main

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git reset --hard origin/main
HEAD is now at 0a5675e Ajout du contrôle de saisie de l'email

Ydrani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$
```

Figure 1: Console git bash avec les résultats des successions de commandes afin de récupérer la dernière version du dépôt

Aussi, il est possible de lire le dernier message de commit : Ajout du contrôle de saisie de l'email.

Scénario test des emails

Un email est normalement de la forme suivante : **chaîneDeCaractères@domaine.aa**

Selon la norme RFC 822, tous les caractères non valides sont :

- les lettres accentuées
- les caractères de contrôle (CTRL + touche clavier)
- les espaces
- les signes spéciaux comme (<>@,;:"'[]|ç%& ,
- Pas de domaine de longueur inférieur à 2

Ainsi, un ensemble de test pourrait être les suivants :

```
assertEquals(true,formulaire.validerEmail("dupont.marie@exemple.fr");
assertEquals(true,formulaire.validerEmail("n-23mE_.l0Ast_9Na-Me@valide01-1234567.frrrrrrrrrrrrrrrrrr"));

assertEquals(false,formulaire.validerEmail("@exemple.fr") ;
assertEquals(false, formulaire.validerEmail("exemple.fr") ;
assertEquals(false, formulaire.validerEmail("n-234Me@inValide[<01-.f") ;
assertEquals(false, formulaire.validerEmail("n-23\\E_.\ "true\ "n-E()_.\ « 0Ast_9-Na4M@inval01-124579.com") ;
```

Mettre à jour le code

Mettre à jour le code actuellement sur le serveur à partir de la dernière version présente dans la « branche » de développement sur GitHub nommée « dev ». Pour ce faire, il faut donc remplacer les fichiers actuels, par ceux présents dans la dernière version de la branche « dev », en utilisant le code suivant : **git reset --hard origin/dev**

```
drani@MSI MINGW64 ~/Programmation/Git/Chocolatein/chocolatein (master)
$ git reset --hard origin/dev
HEAD is now at 172216c Nouvelle fonctionnalité de recherche de produits (beta)
```

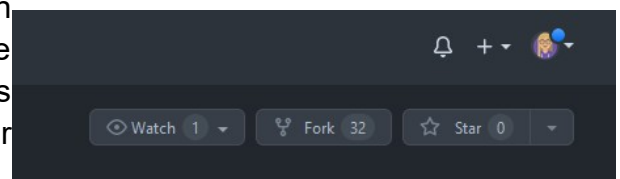
Figure 2: Réponse de la console git bash apres insertion de la commande

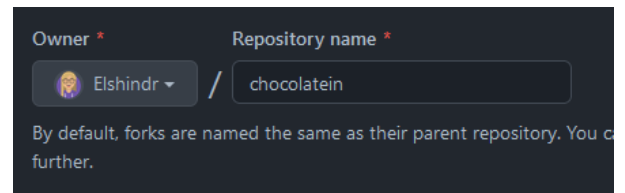
De même, le dernier message de commit de la branche s'affiche : Nouvelle fonctionnalité de recherche de produit.

Détection d'anomalies

Récupération et installation

Afin de fork le projet, il est nécessaire d'avoir un compte Github et de se rendre sur la page du projet en question, [ici](#). C'est bien la branche dev qu'il convient de récupérer. Sur la page en question, il faut cliquer sur le bouton de fork en haut à droite. A la suite, une fenêtre de création s'ouvre afin de créer un dépôt depuis son propre compte lié à ce projet, après avoir saisi un nom, il faut cliquer sur create Fork.

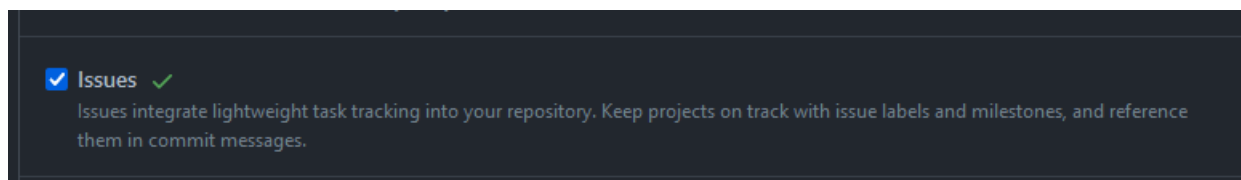




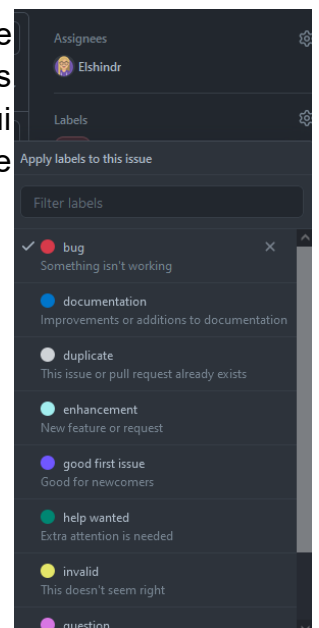
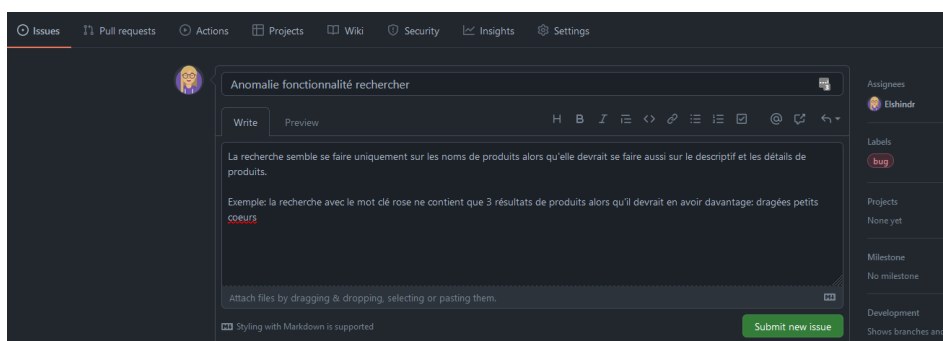
Après avoir Fork le projet, il est possible de récupérer le clone du nouveau dépôt du projet, puis d'installer ce dépôt soit sur un serveur virtuel sous VM, soit via WampServer. Ici, le projet a été glissé-déposé dans la VM de serveur Apache préalablement créé via le logiciel FilaZilla Client dans son dossier **/var/www/html**

Tests de la nouvelle fonctionnalité

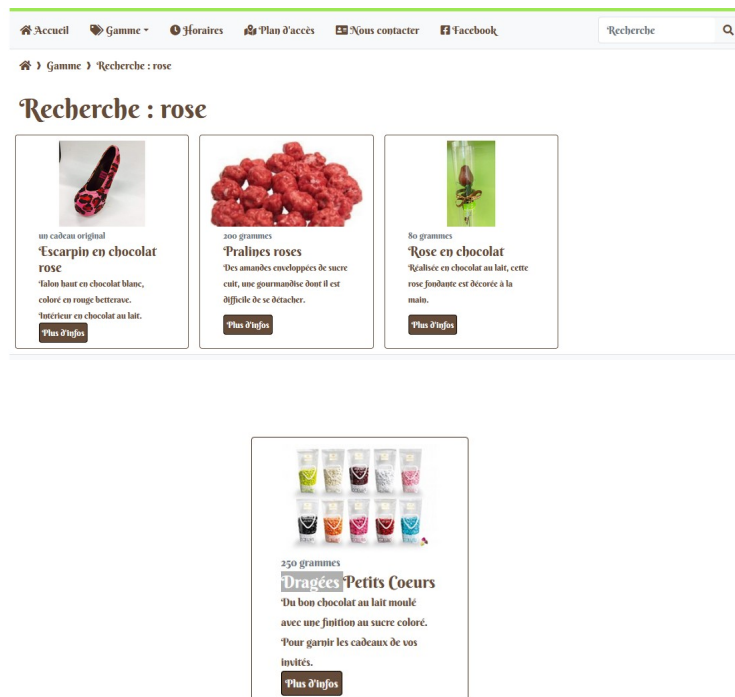
Il est possible d'activer la fonction issue depuis son compte GitHub, dans Settings, Features puis en cochant la case Issues.



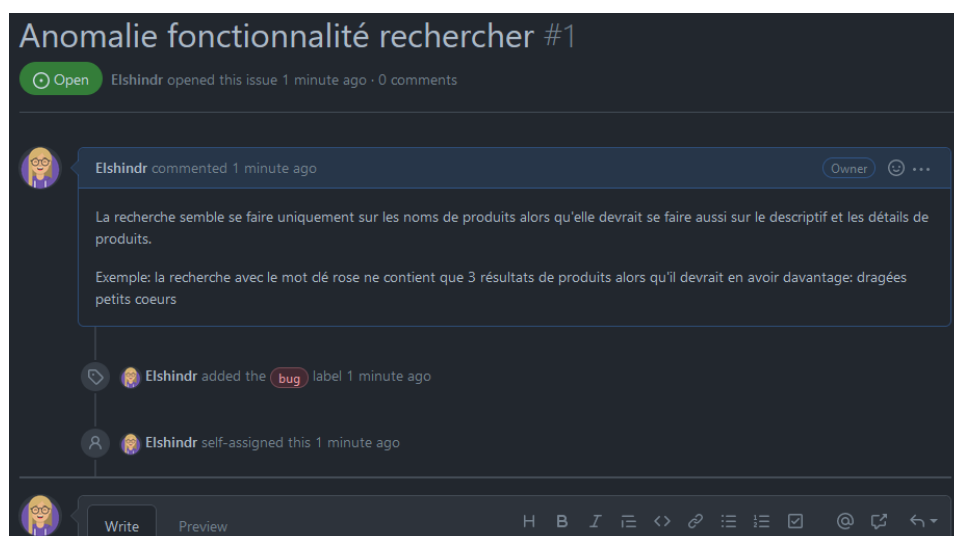
En se rendant dans l'onglet Issue du dépôt distant, un bouton 'New issue' permet de créer une nouvelle Issue en cas de bug détecté. Il est alors possible de reporter ce bug en le décrivant, d'ailleurs il est possible de lui donner un label qui est une étiquette avec une désignation précise permettant d'identifier rapidement le type d'issue, ici il est mis sur bug.



En utilisant la fonction de recherche, il a été possible de constater que la recherche ne se fait que sur les noms de produits et non pas sur sa description ou ses détails. Ce dysfonctionnement sera ainsi reporté. Par exemple, le mot rose ne contient que 3 résultats alors qu'il devrait en avoir plus, comme les « dragées petits coeurs »



Ainsi, après avoir reporté ce bug sur la plateforme Github, il est possible de consulter cette issue à ce [lien](#).



Liste des compétences couvertes

Mettre à disposition des utilisateurs un service informatique :

- Réaliser les tests d'intégration et d'acceptation d'un service
- Accompagner les utilisateurs dans la mise en place d'un service