Table of Contents

Introduction :

Ce mode opératoire permet d'avoir une base de travail pour installer le CRM cremeCrm sur un serveur Ubuntu (debian) issu d'une installation d'OS type serveur.

Maîtriser les particularités pour commencer avec un serveur :

Fixer la variable d'environnement de la langue d' encodage utilisé par le serveur :

```
export LANGUAGE=en_US.UTF-8
export LANG=en_US.UTF-8
export LC_ALL=en_US.UTF-8
locale-gen en_US.UTF-8
dpkg-reconfigure locales

export LANGUAGE=fr_FR.UTF-8
export LANG=fr_FR.UTF-8
export LC_ALL=fr_FR.UTF-8
locale-gen fr fr_FR.UTF-8
```

Disposer des outils d'installation de modules développés en python utilisé par le CRM :

```
sudo apt-get install python-setuptools
sudo apt-get install python-pip
```

Installer les paquets serveur suivants car indispensables à l'installation et l'exploitation du CRM :

```
mysql_config
mysql-python
```

libmysqlclient-dev
python-dev
libxslt1-dev
graphviz
graphviz-dev
hg
apache2
libapache2-mod-python
libapache2-mod-wsgi

Créer les répertoires suivants, l'un pour récupérer et exécuter le source et l'autre pour l'installation :

```
mkdir ~/creme_1.2
mkdir ~/.virtualenvs
```

Ajouter les caractéristiques de session terminal utilisateur suivantes dans le .bashrc :

```
vi ~/.bashrc
```

les caractéristiques de session terminal utilisateur sont :

```
export PYTHONPATH=~/.local
export WORKON_HOME=~/.virtualenvs
export PATH=$PATH:$PYTHONPATH/bin
source ~/.local/bin/virtualenvwrapper.sh
```

Installer les outils de cloisonnement pour les applications en python :

Installer l'outil de cloisonnement :

```
sudo pip install virtualenvwrapper
```

Les commandes de l'outil, par ordre d'arrivé de leur utilisation lors de l'installation, sont :

```
mkvirtualenv
workon
lssitepackages
```

Pour informations :
Des modules divers et variés sont nécessaire à l'installation et l'exploitation du CRM. Afin de permettre une stabilité dans leur inter-indépendance, ces module sont cloisonnés ensemble permettant ainsi de figer ce socle indispensable au CRM.

Installer les modules du CRM

Le répertoire de cloisonnement

Récupérer le source du CRM du dépôt de la dernière version courante stable :

```
cd ~/creme_1.2
hg clone https://bitbucket.org/hybird/creme_crm-1.2
cd ~
```

Construire le répertoire de cloisonnement du socle de modules, répertoire appelé ici *creme* :

```
Source ~/.bashrc
```

mkvirtualenv --no-site-packages creme

Positionner l'utilisateur session terminal sur le répertoire cloisonné et lister ses modules :

```
workon creme
lssitepackages
```

Appliquer les préférences de session terminal utilisateur précedemment paramètrées :

```
source ~/.bashrc
```

Installer le CRM grâce à la gestion des dépendances qu'il sait gèrer via le fichier requirements.txt :

```
cd ~/creme_1.2/creme_crm-1.2/creme
pip install -r requirements.txt
cd ~
```

Attention en cas d'erreur suite à la commande ci dessous, vérifier et faite ce qui suit.

Fixer le répertoire de travail du module graphiz pour notre cas d'installation sur serveur Linux :

```
#Improviser en fonction des informations du message d'erreur
….
```

Si erreur sur graphwiz (binaire introuvable) ;

Vérifiez que graphwiz est bien installé (avec apt-get), puis vérifier que les répertoires suivants existent :

```
ls -al /usr/lib/graphviz
ls -al /usr/include/graphviz
```

Editer le fichier de conf de l'installer Python de graphwiz pour lui dire de chercher manuellement l'emplacement de graphwiz.

```
vi ~/.virtualenvs/creme/build/pygraphviz/setup.py
```

Décommenter aux lignes 39 et 40 les lignes library_path et include_path

```
# Linux, generic UNIX
library_path='/usr/lib/graphviz'
include_path='/usr/include/graphviz'
```

Si erreur sur distribue

Par défaut, en version 1.2, CremeCRM utilise distribue 0.6.15, qui n'est plus disponible.

Il faut juste installer à la place distribute (dernière version) : donc

```
pip installe distribute
```

Installer la base de données du CRM

La base de données

Partant du postulat que l'on utilise le serveur de base de données Mysql

Connecter l'utilisateur en root sur le serveur de base de données pour créer un compte user CRM :

```
mysql -uroot -p
use mysql;
INSERT INTO user(host,user,password) VALUES
('localhost','cremeuser',PASSWORD('ce_que_vous_voulez_pour_password'));
```

Créer la base de données :

```
create database bdcremecrm ;
```

Exécuter la requête suivantes permettant d'accorder tout les droits à l'utilisateur sur la base CRM :

```
GRANT SELECT ,INSERT ,UPDATE ,DELETE ,CREATE ,DROP ,INDEX ,ALTER
,CREATE TEMPORARY TABLES ,CREATE VIEW ,EVENT,TRIGGER,SHOW VIEW
,CREATE ROUTINE,ALTER ROUTINE,EXECUTE ON bdcremecrm.* TO
'cremeuser'@'localhost';
```

Installer le CRM

Configurer le fichier d'installation setting.py (django framework) avec les caractéristiques de votre compte de base données :

```
cd ~/creme_1.2/creme_crm-1.2/creme
vi settings.py
```

les caractéristiques de votre compte de base données sont contenu dans la variable DATABASES :

```
DATABASES = {
    'default': {
        'ENGINE':   'django.db.backends.mysql', # 'postgresql_psycopg2', 'mysql', 'sqlite3' or 'oracle'.
        'NAME':     'bdcremecrm',               # Or path to database file if using sqlite3.
        'USER':     'cremeuser',                # Not used with sqlite3.
        'PASSWORD': 'ce_que_vous_voulez_pour_password',                 # Not used with sqlite3.
        'HOST':     '',                         # Set to empty string for localhost. Not used with sqlite3.
        'PORT':     '',                         # Set to empty string for default. Not used with sqlite3.
        'OPTIONS':   {'init_command': 'SET storage_engine=INNODB' },                    # Extra parameters for
database connection. Consult backend module's document for available keywords.
    },
}
```

Transcrire le modèle objets du CRM en construisant le modèle de données de la base de données

```
cd ~/creme_1.2/creme_crm-1.2/creme
python manage.py syncdb
```

Il arrive qu'au cour de cette transcription, des actions sur la base soit incompatibles avec les contraintes d'intégrités des tables déjà créées au cours de celle-ci. Il faut alors les ajuster en les modifiant de sorte à les rendre compatibles et par la suite relancer la transcription (commande ci dessus)

```
#Necessite de se connecter à la base bdcremecrm
ALTER TABLE `creme_core_blockstate` ADD CONSTRAINT
`creme_core_blockstate_block_id_uniq` UNIQUE (`block_id`) []
```

Créer l'utilisateur et son mot de passe, lorsque l'on vous le proposera :
Cet utilisateur applicatif est un compte de type administrateur CRM

```
cd ~/creme_1.2/creme_crm-1.2/creme
python manage.py migrate --all
```

Peupler la base de données CRM :

```
cd ~/creme_1.2/creme_crm-1.2/creme
python manage.py creme_populate
```

Génère les fichiers web :

```
cd ~/creme_1.2/creme_crm-1.2/creme
python manage.py generatemedia
```

La configuration du serveur Web

Créer le lien symbolique

Pour des raisons de sécurité, j'ai sorti l'environnement crème de l'utilisateur. L'idée c'est que ce soit l'utilisateur d'Apache (www-data) qui l'exécute dans un directory accessible par lui même. J'ai aussi gardé uniquement le sous dossier creme. Donc plus de dossier doc ou des fichiers de dev accessible en production.

L'idée est de mettre le site en production dans un sous dossier de /var/www/, en tant que virtual host.

Déplacer (copie) l'environnement cloisonné sur le dossier de production (/var/www/crm):

```
sudo mkdir /var/www/crm
sudo cp ~/creme_1.2/creme_crm-1.2/creme /var/www/crm/
sudo cp ~/.virtualenvs /var/www/crm/
sudo chown -R www-data:www-data /var/www/crm/
```

Créer le lien symbolique du CRM dans l'environnement cloisonné :

```
cd /var/www/crm
sudo ln -s /var/www/crm/creme/ /var/www/crm/.virtualenvs/creme/lib/python2.7/site-packages/creme
```

Changer le propriétaire et le groupe en www-data (droit du site web) :

```
sudo chown -R www-data:www-data /var/www/crm/
```

Configurer le Wsgi

Ajouter à votre configuration apache et en dehors de votre VirtualHost les lignes suivantes (possible aussi dans le fichier qui décrit le virtual host, mais avant la directive Virtual Host)

```
vi /etc/apache2/sites-available/default
```

Il interprètera ainsi les commandes et déclarations disponible à ces chemins

```
WSGIScriptAlias / /var/www/crm/creme/django.wsgi
WSGIPythonPath /var/www/crm/.virtualenvs/creme/lib/python2.7/site-packages
```

Déclarer votre VirtualHost

Votre VirtualHost pourrait ressembler à celui ci :

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/crm/creme
                                                ...
        <Directory /var/www/crm/creme>
                AllowOverride None
                 Order deny,allow
                 Allow from all
        </Directory>
                                                ...
</VirtualHost>
```

Si vous êtes parano, vous pouvez limiter les requêtes possibles, pour réduire les risques de cracking. Nous autorisons seulement les requêtes GET POST OPTIONS et PROPFIND.

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/crm/creme
                                                ...
        <Directory /var/www/crm/creme>
                AllowOverride None
                 Order deny,allow
                 Allow from all

                <Limit GET POST OPTIONS PROPFIND>
                        Order allow,deny
                        Allow from all
                </Limit>
                <Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK
UNLOCK>
                        Order deny,allow
                        Deny from all
                </Limit>

        </Directory>
                                                ...
</VirtualHost>
```

Configurer le fichier de WSGI

Votre django.wsgi pourrait ressembler à celui ci :

```
import site
import os
import sys

DIR = ['/var/www/crm/creme/']
```

```
from os.path import dirname, join, abspath
CREME_ROOT = dirname(abspath(__file__))

sys.path.insert(2,'/var/www/crm/.virtualenvs')
sys.path.insert(1,'/var/www/crm/.virtualenvs/creme')
sys.path.insert(0,'/var/www/crm/creme')

sys.path.append(CREME_ROOT)


os.environ['DJANGO_SETTINGS_MODULE'] = 'settings'

import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

Configuration supplémentaire

Dans le fichier settings.py, pensez à modifier les lignes suivantes par vos valeurs :

```
SITE_DOMAIN = 'http://monsupersite.com'
MEDIA_URL = 'http://monsupersite.com/media/'
```

Le répertoire de fichiers statiques

Le CRM utilise pour son fonctionnement des fichiers statiques de type *.js, *.png etc.

les fichiers urls.py, settings.py et le fichier de configuration Apache (default par exemple) contiennent respectivement des fonctions, des variables globales et des stratégies de droit d'accès relatives au fichiers statiques.

Exécuter les actions suivantes :

```
#A compléter
```

Le répertoire d'upload de fichiers

Le CRM permet de stocker des fichiers de tous types avec des distinctions suivant leur utilisation prédestinée.

les fichiers urls.py, settings.py et le fichier de configuration Apache (default par exemple) contiennent respectivement des fonctions, des variables globales et des stratégies de droit d'accès relatives au repertoire d'upload de fichiers.

Exécuter les actions suivantes :

```
#A compléter
```

Sécuriser votre serveur Web

Au choix, soit sécurisé par un chiffrement ssl les échanges client/serveur, soit l'accès aux seules personnes disposant d'un mot de passe htaccess.

Configurer le SSL :

```
#Simplement créer un virtual host SSL, puis copier coller le virtual host dedans en remplaçant
<VirtualHost *:80>  par <VirtualHost *:443>
```

Configurer l'htaccess :

Dans la directive Apache, il faut autoriser l'usage d'un .htaccess.

Dans le virtual Host, il faut remplacer la ligne AllowOverride None par AllowOverride All

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost

        DocumentRoot /var/www/crm/creme
                                                ...
        <Directory /var/www/crm/creme>
                AllowOverride All
                 Order deny,allow
                 Allow from all
        </Directory>
                                                ...
</VirtualHost>
```

Ensuite, il faut créer un fichier .htpasswd. Il faut remplacer username par le nom de la personne : fred par exemple.

```
#sudo htpasswd -c /etc/apache2/.htpasswd username
```

Après, créer un fichier .htaccess à la racine du site :

```
#sudo vi /var/www/crm/creme/.htaccess
```

Et le compléter comme ceci :

```
AuthUserFile /etc/apache2/.htpasswd
AuthGroupFile /dev/null
AuthName "Authetfication obligatoire"
AuthType Basic

<Limit GET POST>
order deny,allow
deny from all
require valid-user
satisfy any
</Limit>


<Files .htaccess>
 order allow,deny
 deny from all
</Files>
```

Et enfin, redémarrer Apache :

```
#sudo service apache2 restart
```