

Mise en place de l'environnement de production

Objectif

Cette documentation va expliquer comment mettre en place un environnement de production avec un serveur Debian, configurer pour exécuter le service Apache chargé de servir le framework Web2Py.

Pre-requis

- Une ferme de serveur avec assez de ressource disponible
- Le vSphere Client de VMWare
- L'ISO de Debian 7 *netinst*

Création de la VM

Premièrement, nous allons nous connecter à la ferme de serveur grâce à vSphere Client avec nos identifiants fournis en début d'année.



Lorsque l'on va cliquer sur le bouton **Connexion** nous allons obtenir un avertissement de sécurité. L'avertissement est normal puisque le certificat utilisé est auto-généré par le serveur. On se retrouve sur l'écran d'accueil.



Nous lançons la procédure pour créer une nouvelle machine virtuelle en cliquant sur *Fichier > Créer une nouvelle machine virtuelle*

1. Choisir une configuration **Typique**
2. Saisir le nom de notre machine **M2LWebProd** que l'on place dans *VMS Etudiants > PPE2 > M2L > Groupe 1*
3. Dans la partie *Hôte/cluster*, choisir **CLUSTER**
4. Mettre la VM dans le *Pool Pédagogique*
5. Sélectionner l'espace de stockage **SLAM**
6. Le système d'exploitation sera un *Linux*, version *Debian GNU/Linux 6 (64 bits)*
7. La VM se trouvera sur le réseau **VLAN501**
8. Créer un disque dur de 5Go en provisionnement dynamique.

Après avoir complété les huit étapes, on arrive sur un écran de résumer comme suit :



Il ne reste plus qu'à valider la fenêtre pour lancer la création de la VM.

Installation de Debian 7

✘ Pour commencer l'installation, nous devons commencer par monter l'ISO télécharger comme indiquer dans les [pre-requis](#).

Pour y parvenir, nous démarrons la VM et cliquons sur l'icône en forme de CD avec une clé, on sélectionne le lecteur de la VM *Lecteur CD/DVD 1* puis *Connecter l'image ISO sur le disque local...* Une fenêtre demandant d'indiquer le chemin vers l'ISO s'ouvre. On lui indique, valide. On se rend ensuite dans la console et on exécute le raccourci clavier : **Ctrl + Alt + Insert** pour redémarrer la VM.

Lors du démarrage sur l'image ISO debian affiche un menu avec différents choix on sélectionne **Install** A la suite de cette étape l'installateur va nous demander différents éléments de configuration dans cette ordre :

- Sélection de la langue : ici français
- Choix de la location géographique : ici France
- Choix du layout(disposition) du clavier : ici français
- Configuration auto du réseau

On obtient une erreur DHCP, cette erreur intervient car aucun serveur DHCP n'est présent sur le réseau



- Configuration manuelle du réseau : ici on utilise l'IP fournit
- Configuration du masque : ici *255.255.255.0*
- Configuration de la passerelle : ici *172.21.101.254*
- Configuration de l'IP du serveur de nom : ici *172.21.101.254*
- Configuration du nom d'hôte de la machine : ici *M2LWEBProd*
- Choix du mot de passe root : ici on ne n'en met pas pour désactiver le root
- Choix d'un nom d'utilisateur
- Choix du mot de passe + confirmation
- Configuration du partitionnement du disque dur : ici on met tout sur la même partition



- Configuration de l'adresse du miroir : ici *ftp.fr.debian.org*
- Configuration de l'adresse du mandataire : ici <http://192.168.222.85:3142/>
- Demande de participation a une étude statistique : ici on répond *Non*
- Sélection des logiciels : ici on ne coche que *Utilitaires usuels du système*



- Installation de GRUB : ici on répond *Oui*

L'installation est maintenant terminée, on peut éjecter l'ISO en utilisant la même procédure que celle utilisée au début de cette partie.

Mise en place des outils

On commence avec l'installation du serveur **OpenBSD** (ssh):

```
# apt-get install -y ssh
```

Ensuite on passe à l'installation d'**Apache 2** :

```
# apt-get install apache2
```

Puis au déploiement du framework **web2py** sur la machine à l'aide de la [documentation](#) : On installe les paquets zip et unzip

```
# apt-get -y install zip unzip
```

On configure le proxy pour wget

```
# vi /etc/wgetrc
79 https_proxy = http://172.30.137.29:3128/
80 http_proxy = http://172.30.137.29:3128/
81 ftp_proxy = http://172.30.137.29:3128/
```

On télécharge l'archive et on l'extrait

```
# wget http://web2py.com/examples/static/web2py_src.zip
# unzip web2py_src.zip
```

On déplace le fichier **wsgihandler.py** permettant l'interfaçage entre web2py et apache et on change le propriétaire du dossier pour **www-data**

```
# mv web2py/handlers/wsgihandler.py web2py/wsgihandler.py
# chown -R www-data:www-data web2py
```

On configure Apache, sans oublier d'installer le module wsgi

```
# apt-get -y install libapache2-mod-wsgi
# a2enmod ssl
# a2enmod proxy
# a2enmod proxy_http
# a2enmod headers
# a2enmod expires
# a2enmod wsgi
# a2enmod rewrite
# service apache2 restart
```

On crée ensuite un certificat auto signé pour accéder à l'administration de web2py

```
# mkdir /etc/apache2/ssl
# openssl genrsa 1024 > /etc/apache2/ssl/self_signed.key
```

```
# chmod 400 /etc/apache2/ssl/self_signed.key
# openssl req -new -x509 -nodes -sha1 -days 365 -key
/etc/apache2/ssl/self_signed.key > /etc/apache2/ssl/self_signed.cert
# openssl x509 -noout -fingerprint -text < /etc/apache2/ssl/self_signed.cert
> /etc/apache2/ssl/self_signed.info
```

On installe la configuration pour Apache et on l'active

```
# vi /etc/apache2/sites-available/web2py
<VirtualHost *:80>
  WSGIDaemonProcess web2py user=www-data group=www-data display-
name=%{GROUP}
  WSGIProcessGroup web2py
  WSGIScriptAlias / /var/www/web2py/wsgihandler.py

  <Directory /var/www/web2py>
    AllowOverride None
    Order Allow,Deny
    Deny from all
    <Files wsgihandler.py>
      Allow from all
    </Files>
  </Directory>

  AliasMatch ^/([^/]+)/static/(.*)
/var/www/web2py/applications/$1/static/$2
  <Directory /var/www/web2py/applications/*/static/>
    Order Allow,Deny
    Allow from all
  </Directory>

  <Location /admin>
    Deny from all
  </Location>

  <LocationMatch ^/([^/]+)/appadmin>
    Deny from all
  </LocationMatch>

  CustomLog /var/log/apache2/access.log common
  ErrorLog /var/log/apache2/error.log

</VirtualHost>

<VirtualHost *:443>
  SSLEngine on
  SSLCertificateFile /etc/apache2/ssl/self_signed.cert
  SSLCertificateKeyFile /etc/apache2/ssl/self_signed.key

  WSGIProcessGroup web2py
```

```
WSGIScriptAlias / /var/www/web2py/wsgihandler.py

<Directory /var/www/web2py>
  AllowOverride None
  Order Allow,Deny
  Deny from all
  <Files wsgihandler.py>
    Allow from all
  </Files>
</Directory>

AliasMatch ^/([^/]+)/static/(.*)
/var/www/web2py/applications/$1/static/$2

<Directory /var/www/web2py/applications/*/static/>
  Order Allow,Deny
  Allow from all
</Directory>

CustomLog /var/log/apache2/access.log common
ErrorLog /var/log/apache2/error.log

</VirtualHost>

# a2ensite web2py
# service apache2 restart
```

On définit ensuite le mot de passe administrateur avec les commandes :

```
# sudo -u www-data python -c "from gluon.widget import console; console();"
# sudo -u www-data python -c "from gluon.main import save_password;
save_password(raw_input('admin password: '),443)"
```

Après cette installation, nous avons rencontré un problème de réécriture d'URL avec Apache. Le numéro de version apparaissait après **/static/** dans l'URL des fichiers de l'application **admin**. Pour corriger le problème, nous avons remplacé les deux lignes du fichier `/etc/apache2/sites-available/web2py`

```
AliasMatch ^/([^/]+)/static/(.*) /var/www/web2py/applications/$1/static/$2
```

par

```
AliasMatch ^/([^/]+)/static/(?[_\d]+\.[\d]+\.[\d]+)/?(.*)
/var/www/web2py/applications/$1/static/$2
```

From:

<https://wiki.viper61.fr/> - **Viper61's Wiki**

Permanent link:

https://wiki.viper61.fr/sio/ppe2/mission_3/env_prod?rev=1426090622

Last update: **18/09/2016 02:54**