

Installation de ProWhy >= 1.3 sous Linux

Anciennes versions :

| Pour installer la version 1.0, c'est ici : [\[\[InstallLinuxProwhy1.0|Installation Prowhy 1.0\]\]](#)

Mise à jour :

| Pour mettre à jour ProWhy < 1.3 vers ProWhy 1.3, suivre la documentation ici : [\[\[UpgradeProwhy1.3|Upgrade vers ProWhy 1.3\]\]](#)

ProWhy doit être installé sur un serveur.
C'est une application web développée avec le framework Ruby On Rails qui peut être installée sur n'importe quel serveur Linux. L'installation sur serveur Windows est également possible mais non documentée.

La documentation complète d'installation sous Linux est disponible sur ce site.

L'utilisateur n'a rien à installer sur sa machine et peut utiliser n'importe quel système d'exploitation (windows, linux, ios, android etc).

Il lui faut seulement un navigateur récent et une connexion au serveur.

L'application est optimisée pour Mozilla Firefox, mais peut également être utilisée depuis IE, Chrome, Safari, etc.

Méthode alternative :

| Nous mettons à votre disposition une **machine virtuelle Debian/ProWhy** installée et configurée afin de permettre le test et/ou l'utilisation de l'application ProWhy sans avoir à réaliser l'installation complète du serveur. Vous pouvez ainsi installer le serveur ProWhy sur une machine **Windows, Linux, Mac OSX ou Solaris**

Téléchargement et documentation : [VM Debian / ProWhy](#)

En cas de difficultés avec l'installation, vous pouvez poser vos questions dans le forum :

[Installation et configuration](#)

Installation testée sous Ubuntu 14.04 / 16.04 et Debian 8 Jessie

1. Installation de Ruby.

Installations nécessaires pour rvm, ruby, rails, etc :

```
$ sudo apt-get update
```

```
$ sudo apt-get install git-core curl zlib1g-dev build-essential libreadline-dev libxslt-dev libxslt1-dev libxml2 libxml2-dev openssl libssl-dev libyaml-dev libsqlite3-dev sqlite3 autoconf libc6-dev ncurses-dev automake libtool bison subversion python
```

Installation de rvm (Ruby Versions Manager).

Rvm permet l'installation et la gestion de plusieurs versions de Ruby sur la même machine. Cette installation permettra une évolution simplifiée vers les versions futures de ProWhy.

Il est préférable d'effectuer **l'installation rvm sans les droits d'administrateur**, rvm sera installé pour l'utilisateur courant. Si vous souhaitez que l'installation rvm soit accessible pour tous les utilisateurs, vous devez ajouter sudo devant "bash" et ajouter la commande **rvmsudo** devant les chargements de gems.

Installation de rvm (=> informations dans <https://rvm.io/rvm/install>)

- Importer la clé :
\$ \curl -sSL https://rvm.io/mpapis.asc | gpg --import -

- Chargement et installation de rvm
\$ \curl -sSL https://get.rvm.io | bash -s stable

- Ajouter à la fin du fichier ~/.bashrc :

```
export PATH="$PATH:$HOME/.rvm/bin" # Add RVM to PATH for scripting
[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm" # Load RVM into a shell session *as a function*
```

Attention : si l'installation a été effectuée en root, remplacer "\$HOME/.rvm" par "/usr/local/rvm"

- Exécuter :
\$ source ~/.bashrc

Installation de Ruby 1.9.3

- Configuration du proxy
Si vous utilisez un proxy, il est possible que ruby ne se charge pas, auquel cas vous devez configurer le proxy dans git avec les commandes suivantes :

```
$ export http_proxy=http://user:pass@proxyhost:proxyport
$ git config --global http.proxy http://user:pass@proxyhost:proxyport
```

- Chargement et installation de ruby :
\$ rvm install 1.9.3-p551
\$ rvm use 1.9.3
\$ rvm --default use 1.9.3

- En cas de problème avec l'installation :
\$ rvm requirements
Si besoin installer les composants nécessaires, et relancer la commande.

Attention : pour lancer une application rvm en sudo, utiliser rvmsudo

2. Installations supplémentaires.

MySql

```
$ sudo apt-get install mysql-client mysql-server
$ sudo apt-get install libmysqlclient-dev ruby-mysql
```

Avec Debian 9 (Stretch) remplacer la dernière ligne par :

```
$ sudo apt-get install default-libmysqlclient-dev
$ sudo apt-get install ruby-mysql2
```

Installation de la librairie RMagick.

```
$ sudo apt-get install libmagickcore-dev libmagickwand-dev
```

Installation de wkhtmltopdf.

Installer une **version ≥ 0.12.0**

Pour Debian Jessie 64bits :

```
$ wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.2.1/wkhtmltox-0.12.2.1_linux-jessie-amd64.deb
$ sudo dpkg -i wkhtmltox-0.12.2.1_linux-jessie-amd64.deb
```

Pour Debian Jessie 32bits :

```
$ wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.2.1/wkhtmltox-0.12.2.1_linux-jessie-i386.deb
$ sudo dpkg -i wkhtmltox-0.12.2.1_linux-jessie-i386.deb
```

Pour les autres distributions, installer une version ≥ 0.12.0 depuis la page

<https://github.com/wkhtmltopdf/wkhtmltopdf/releases/0.12.2.1/> ou <http://wkhtmltopdf.org/downloads.html>

Tester :

```
$ wkhtmltopdf http://www.google.com output.pdf
=> ouvrir le fichier output.pdf
```

3. Chargement et test de l'application.

Chargement avec svn (subversion).

- Créer un répertoire prowhy et charger l'application

```
$ mkdir prowhy
$ svn checkout http://www.prowhy.org/svn/prowhy/stable/prowhy prowhy
```

*Cette méthode vous permet d'avoir accès aux mises à jour
Pour faire la mise à jour, dans le dossier prowhy :*

Récupération des évolutions de la version :

```
$ svn update .
```

Clear des sessions

```
$ rake db:sessions:clear
$ rake db:tmp:clear
$ rake tmp:clear
```

Vérification des éventuelles évolutions de base de données à effectuer :
Vérifier sur la page "Annonces"
Relancer le serveur apache

```
$ sudo service apache2 restart
```

Installation des composants nécessaires à l'application (gems).

dans le répertoire de l'application (prowhy)

```
$ cd prowhy (cela va créer le gemset pour rvm associé à l'application)
$ bundle install
```

si message d'erreur :
bundle : commande introuvable
lancer la commande :
\$ gem install bundler
puis relancer la commande
\$ bundle install

Mise en place de la BDD.

Dans le répertoire de l'application (prowhy)

- Éditer le fichier config/database.yml :
 - si besoin, modifier les noms des bases de données en mode développement et production
 - modifier le mot de passe d'accès à mysql
- Création de la database :

```
$ rake db:create
```

- Création des tables :
\$ rake db:migrate

Si vous avez une erreur du type :
migrate Mysql2::Error: All parts of a PRIMARY KEY must be NOT NULL; if you need NULL in a key, use UNIQUE instead:

Créez le fichier : config/initializers/abstract_mysql2_adapter.rb

Contenant :

```
require 'active_record/connection_adapters/mysql2_adapter'  
class ActiveRecord::ConnectionAdapters::Mysql2Adapter  
  NATIVE_DATABASE_TYPES[:primary_key] = "int(11) auto_increment PRIMARY KEY"  
end
```

Et relancez la commande précédente.

- Insertion des données :
\$ rake db:fixtures:load

Tester l'application : dans le répertoire de l'application (prowhy)

\$ rails s

Accéder à l'application depuis un navigateur à l'adresse localhost:3000

Accéder à l'application avec un des logins suivants :

- Administrateur

login : admin
passwd : admin

- Utilisateur

login : user1
passwd : user1

4. Déploiement avec Apache en mode production.

Installer Apache2

\$ sudo apt-get install apache2

Installer le module passenger pour Apache

\$ gem install passenger -v 4.0.58 --no-rdoc --no-ri

Attention : il semblerait qu'il y ait des problèmes de fonctionnement avec les versions récentes de passenger aussi ne pas oublier de mettre la version 4.0.58

\$ rvmsudo passenger-install-apache2-module

Cette commande vérifie de nombreuses dépendances.

Si besoin, effectuer les installations demandées et relancer passenger-install-apache2-module

L'installateur vous indique les chemins du module passenger à reporter dans la création du module ci-dessous.

Chargement du module passenger dans Apache

Créer 2 fichiers passenger.load et passenger.conf dans /etc/apache2/mods-available (adapter les chemins en fonction de votre configuration)

passenger.load :

```
LoadModule passenger_module
/home/prowhy/.rvm/gems/ruby-1.9.3-p551@prowhy-rails-3.2/gems/passenger-4.0.58/buildout/apache2/mod_passenger.so
```

passenger.conf

```
<IfModule mod_passenger.c>
  PassengerRoot /home/prowhy/.rvm/gems/ruby-1.9.3-p551@prowhy-rails-3.2/gems/passenger-4.0.58
  PassengerDefaultRuby /home/prowhy/.rvm/gems/ruby-1.9.3-p551@prowhy-rails-3.2/wrappers/ruby
</IfModule>
```

Activer le module passanger pour Apache :
\$ sudo a2enmod passenger

Création de l'utilisateur "passenger" et droits d'accès

```
$ sudo useradd --system passenger
```

Ajoutez votre utilisateur courant dans le groupe passanger afin d'avoir toujours les droits pour effectuer des modifications et les mises à jour de l'application (l'information n'est prise en compte par le système qu'après redémarrage):

```
$ sudo usermod -a -G passenger USER_NAME
```

Ajouter l'information dans la configuration de Passenger (dans apache2/mods-availability/passenger.conf) :

```
PassengerDefaultUser passenger
```

Attribuer les droits d'accès pour l'utilisateur passanger :

Attention : **l'application doit pouvoir écrire dans les répertoires et sous répertoires de public/pdf, public/docs et public/images/indicateurs**

(adaptez le chemin en fonction de votre installation)

```
$ sudo chown passenger:passenger ~/prowhy/config/environment.rb
```

```
$ sudo chmod 664 ~/prowhy/config/environment.rb
```

```
$ sudo chown -R passenger:passenger ~/prowhy/public
```

```
$ sudo chmod -R 775 ~/prowhy/public
```

Création du lien vers le dossier public de l'application (dans /var/www par exemple)

```
$ sudo ln -s ~/prowhy/public /var/www/prowhy
```

Paramétrage : passage de ProWhy en mode production

Création de la database production :

```
$ rake db:create RAILS_ENV=production
```

```
$ rake db:migrate RAILS_ENV=production
```

```
$ rake db:fixtures:load RAILS_ENV=production
```

Configuration du Virtual Host Apache en mode production

Dans cet exemple, nous conservons la configuration par défaut d'Apache, et nous ajoutons notre configuration de prowhy telle que l'accès à l'application soit réalisé avec : http://adresse_du_serveur/prowhy

- Vérifier que le fichier de configuration par défaut (/etc/apache2/sites-enabled/000-default.conf) contient la ligne suivante :

```
DocumentRoot /var/www
# Si le chemin est différent, le changer
```

- Créer le fichier /etc/apache2/sites-available/prowhy_vhost.conf :

```
<Location /prowhy>
```

```
# mode production :  
RailsEnv production
```

```
# Si plusieurs versions de Ruby ont été installées avec rvm, cette ligne permet de définir  
# le chemin de la version à utiliser, si elle est différente de la configuration de Passenger.  
# Dans ce cas, décommenter la ligne suivante et mettre le chemin relatif à votre installation  
# PassengerRuby /home/prowhy/.rvm/gems/ruby-1.9.3-p551@prowhy-rails-3.2/wrappers/ruby
```

```
RackBaseURI /prowhy  
Options -MultiViews  
</Location>
```

- Activer le site prowhy dans Apache :
\$ sudo a2ensite prowhy_vhost

- Relancer le serveur Apache
\$ sudo service apache2 restart

Si apache ne démarre pas correctement (Apache version 2.4), il est nécessaire de modifier l'activation par défaut de certains modules :

```
$ sudo a2dismod mpm_event  
$ sudo a2enmod mpm_prefork  
$ sudo service apache2 restart
```

Vous pouvez désormais accéder à l'application prowhy :
http://adresse_du_serveur/prowhy

Login : admin
Passwd : admin