

# Installation

## Contents

<b>1</b>	<b>Installation d'Emacs</b>	<b>1</b>
1.1	Version proposée par la distribution . . . . .	1
1.2	Préparation de l'installation . . . . .	1
1.3	Nettoyage du système. . . . .	2
<b>2</b>	<b>Récupérer la configuration</b>	<b>2</b>
2.1	Sauvegarde de l'ancienne configuration . . . . .	2
2.2	Téléchargement de la configuration . . . . .	2
<b>3</b>	<b>Python</b>	<b>3</b>
<b>4</b>	<b>Global</b>	<b>3</b>
<b>5</b>	<b>C++</b>	<b>3</b>
<b>6</b>	<b>Chargement accéléré</b>	<b>4</b>
<b>7</b>	<b>Mise à jour</b>	<b>4</b>

Ce document se place dans le cas d'un système **GNU/Linux** pour une distribution **Debian** (ou basée sur **Debian** comme **Ubuntu** par exemple) et présuppose que l'utilisateur peut installer les logiciels par le biais de la commande **apt** et dispose donc des privilèges **root**.

Dans le cas contraire, il reste possible d'utiliser cette configuration, mais l'utilisateur devra, soit utiliser les commandes *ad hoc* (dans le cas d'une autre distribution) soit les installer en recompilant éventuellement les logiciels.

### IMPORTANT

Au premier lancement d'Emacs, tous les paquets nécessaires seront téléchargés puis compilés. Le premier démarrage prendra donc un certain temps. Ce ne sera évidemment plus le cas par la suite.

## 1 Installation d'Emacs

Pour que cette configuration fonctionne de manière optimale il faut que la version d'Emacs installée soit au moins la 26.0.50.

Pour commencer il faut vérifier si le système fourni de la version minimum nécessaire. Dans le cas contraire, il faut enrichir la base des paquets ou l'installer manuellement.

### 1.1 Version proposée par la distribution

Pour connaître la version du système on utilise la commande

```
lsb_release -a
```

Jusqu'aux versions

- **bionic** (18.x.x) d'**Ubuntu**,
- **stretch** (9.x.x) de **Debian**,

incluses, la version d'Emacs n'est pas suffisante.

Si **Emacs** n'est pas installé et si les versions d'**Ubuntu** ou **Debian** sont suffisamment récentes, il suffit d'exécuter dans ce cas :

```
sudo apt update
sudo apt -y install emacs
```

Si Emacs était installé, on peut tout de même s'assurer de sa version en utilisant la commande.

```
emacs --version
```

Si la version est suffisante (au delà de 26.0.50), on peut ignorer le paragraphe suivant.

## 1.2 Préparation de l'installation

Pour installer Emacs manuellement (ce qui n'est pas recommandé), on peut suivre les instructions données ici.

On se concentre ici sur une installation de paquets précompilés.

On doit ajouter des dépôts de paquets pour l'installation, on s'assure donc que les outils de gestion de dépôts extérieurs sont disponibles.

```
sudo apt -y install software-properties-common
```

On ajoute le dépôt en faisant.

```
sudo add-apt-repository -y ppa:kellejk/emacs
```

Puis on installe Emacs.

```
sudo apt -y install emacs26
```

Pour terminer on s'assure que la version par défaut d'Emacs sera bien la 26. Pour cela, on exécute

```
sudo update-alternatives --set emacs /usr/bin/emacs26
```

On peut vérifier à nouveau que tout s'est bien passé en exécutant

```
emacs --version
```

## 1.3 Nettoyage du système.

Si une version d'Emacs était déjà installée, il est probable que des paquets Emacs l'aient été aussi. Pour que cette configuration fonctionne de manière optimale, il convient de les supprimer.

```
sudo apt remove -y --purge '^elpa*'
```

# 2 Récupérer la configuration

## 2.1 Sauvegarde de l'ancienne configuration

Si Emacs avait déjà été utilisé sur le compte de l'utilisateur, il a généré automatiquement des fichiers de configuration. Il faut donc les supprimer (ou les sauvegarder) avant d'installer cette configuration.

Ce sont des fichiers cachés que nous allons placer dans un répertoire pour éventuellement les réutiliser ultérieurement si cette configuration ne convient pas.

```
mkdir -p ~/sauvegarde.emacs-`date +%Y%m%d`
mv ~/.emacs* ~/sauvegarde.emacs-`date +%Y%m%d`
```

## 2.2 Téléchargement de la configuration

Pour obtenir la configuration, il est préférable d'utiliser `git`, pour l'installer il suffit d'exécuter la commande suivante.

```
sudo apt -y install git
```

On peut maintenant récupérer la configuration, soit en si on dispose d'un compte sur `gitlab.delpinix.fr` et qu'on y a déposé une clé `ssh` pour s'authentifier :

```
git clone git@gitlab.delpinux.fr:delpino/dot-emacs.git ~/.emacs.d
```

ou alors de manière anonyme

```
git clone https://gitlab.delpinux.fr/delpino/dot-emacs.git ~/.emacs.d
```

Il est maintenant possible de lancer Emacs. Comme on l'a dit, la configuration initiale va prendre un peu de temps. Et Emacs lui même va installer des paquets (Debian ou Ubuntu) pour satisfaire certaines dépendances.

Durant le processus d'installation des paquets, Emacs posera quelques questions simples (et éventuellement entrer son mot de passe pour les interactions `sudo`). En cas de problème, relancer Emacs et répondre à nouveau aux questions.

Une fois le premier lancement terminé, on fermera Emacs, puis on pourra éventuellement le relancer.

Pour que toute la configuration soit totalement opérationnelle, il faut cependant installer quelques outils supplémentaires.

### 3 Python

Pour utiliser correctement Python avec cette configuration, il faut installer Python (évidemment) et `virtualenv` qui va permettre d'exécuter les scripts directement dans Emacs et bénéficier de l'analyse syntaxique à la volée, entre autres.

```
sudo apt -y install python virtualenv python3-jedi python-jedi
```

Il reste une dernière étape à accomplir, à savoir configurer le serveur `jedi` dans Emacs.

Pour cela, on lance Emacs puis la commande `jedi:install-server`, à savoir dans Emacs taper

où `M-x` signifie qu'on appuie simultanément sur les touches `ALT` et `x`, on tape ensuite `jedi:install-server` à l'invite du *minibuffer* (tout en bas).

Ensuite à la prochaine édition d'un fichier Python, Emacs terminera la configuration en demandant (dans le *minibuffer*)

```
Please answer y or n. Automatically install the RPC dependencies
from PyPI (needed for completion, autoformatting and documentation) ?
(y or n)
```

Il convient évidemment de répondre `y` à cette question.

### 4 Global

On recommande aussi d'installer les `ggtags`.

```
sudo apt -y install global
```

### 5 C++

Enfin, pour configurer correctement l'environnement de développement pour le C++, il faut effectuer quelques installations et configurations supplémentaires.

On commence donc par installer `clang-10`. Il faut s'assurer que le dépôt contenant fait partie de la liste des dépôts. Pour les systèmes Ubuntu, exécuter

```
sudo apt-add-repository -u "deb http://archive.ubuntu.com/ubuntu/ $(lsb_release -cs)-backports main re
```

Pour les systèmes Debian

```
sudo apt-add-repository -u "deb http://deb.debian.org/debian $(lsb_release -cs)-backports main contri
```

On peut alors lancer l'installation proprement dite.

```
sudo apt -y install clang-10 clangd-10 libclang-10-dev clang-format-10
```

On va maintenant aider Emacs à trouver ces bonnes versions de compilateur. Pour cela, on va exécuter les commandes suivantes.

```
mkdir -p ~/bin
ln -s /usr/bin/clang++-10 ~/bin/clang++
ln -s /usr/bin/clang-10 ~/bin/clang
ln -s /usr/bin/clang-format-10 ~/bin/clang-format
```

Il faut alors ajouter

```
export PATH=~/bin:$PATH
```

à son `~/bashrc` si ce n'est pas déjà fait. Si on utilise un autre *shell*, on modifiera la variable `PATH` de la manière adéquate pour que `~/bin` apparaisse au début.

## 6 Chargement accéléré

Après s'être assuré assuré que la configuration fonctionne correctement, on peut éventuellement améliorer le confort d'utilisation. En effet, comme cette configuration est un peu lourde, le temps de lancement d'Emacs peut être un peu long.

Le temps de chargement peut être fortement réduit en lançant Emacs en mode client (plusieurs client s'attachent à un même serveur). Pour bénéficier de cette optimisation ajouter les alias suivants à votre *shell*

```
alias emacs="emacsclient -q -a '' -c"
alias kill-emacs="emacsclient -q -e '(let ((last-nonmenu-event nil))(save-buffers-kill-emacs))'"
```

La première ligne indique de lancer Emacs en mode client si possible. Si il n'y a pas de serveur, il est créé en toute transparence.

## 7 Mise à jour

Pour mettre à jour la configuration, ne suffit pas (en général) de faire un simple `git pull`. La raison est que la base des paquets utilisée évolue et certains doivent éventuellement être mis à jour.

```
\emacs --execute "(progn\
  (try-and-refresh 'auto-package-update)\
  (auto-package-update-now)\
  (message '\"Calling 'git pull...'\")\
  (when (shell-command \"cd ~/.emacs.d; git pull\")\
    (save-buffers-kill-emacs)))"
```

Si tout se déroule correctement, tous les clients Emacs ouverts seront fermés ainsi que le serveur. Le lancement suivant d'Emacs terminera la mise à jour en installant éventuellement de nouveaux paquets. Si des processus sont actifs, Emacs demandera confirmation avant sa fermeture.