



# Setup Ubuntu #1

▼ Version	V1
☰ Type	Technique
📅 Date de création	@14 décembre 2023 15:16
📅 Dernière modification	@14 décembre 2023 15:22

## 🚀 Premier démarrage Ubuntu

Après avoir terminé l'installation de votre distribution Ubuntu et des paquets pré-requis.



Les termes “application” ou “logiciel” sont plus courants et conviviaux, tandis que “paquet” est un terme plus technique qui englobe l'ensemble des éléments nécessaires au fonctionnement d'une application. Le choix du terme est à déterminer en fonction du public mais ces deux termes désignent la même chose.



Apt est un outil en ligne de commande (CLI : Command Line Interface) utilisé pour gérer les paquets sur les systèmes basés sur Debian, tels qu'Ubuntu. Les paquets installés via apt tirent parti des bibliothèques et des dépendances déjà présentes sur le système.

Ubuntu Software est une interface graphique fournie par Ubuntu, destinée à la gestion des logiciels au format Snap. Contrairement aux paquets installés via apt, les paquets au format Snap sont encapsulés avec leurs bibliothèques et dépendances dans des conteneurs isolés.

En conclusion, utiliser apt pour une approche système native et Snap pour une gestion flexible et sécurisée.

### 1. Faire les mises à jour des paquets installés avec apt.

Télécharger les mises à jour disponibles.

```
sudo apt update
```

Installer les mises à jour téléchargées.

```
sudo apt upgrade
```

### 2. Faire les mises à jour des paquets installés avec Snap.

- ☐ Lancer l'application Ubuntu software puis se rendre dans l'onglet "Updates".
- ☐ Exécuter les mises à jour des paquets.
- ☐ Exécuter les mises à jour de votre système si nécessaires ( à faire après la mise à jour des applications ).

Vous pouvez aussi faire les mises à jours à l'aide de la commande suivante :

```
sudo snap refresh snap-store
```

## Pré-requis

Depuis un terminal exécuter les commandes suivantes, ces paquets seront nécessaires pour les installations suivantes.

### 1. Installer Git

```
sudo apt install git
```

### 2. Installer Curl

```
sudo apt install curl
```

## Installations et configurations

### Google Chrome

 [Page de téléchargement de Google Chrome](#)

- ☐ Télécharger la version de Google Chrome adaptée.
- ☐ Cliquez droit sur le fichier téléchargé puis sélectionnez `Open with other application`.
- ☐ Choisir `Software install` parmi la liste des applications recommandées, cela ouvrira une fenêtre Ubuntu Software.

- ☐ Cliquer sur installer en haut à droite.

## Terminator

 [Documentation Ubuntu pour Terminator](#)

Installer Terminator en cliquant sur le lien du chapitre 2.1 de la documentation ou en exécutant la commande suivante :

```
sudo apt install terminator
```

- ☐ Cliquez droit dans une page de Terminator puis sélectionner "Preferences", rendez-vous dans l'onglet "Keybindings".
- ☐ Changer les raccourcis du copier / coller par `Ctrl + c` et `Ctrl + v`.

## Configurer le terminal

### 1. Installer zsh

 [Documentation Ubuntu pour zsh](#)



Le shell est un programme qui agit comme une interface en ligne de commande ( CLI ) entre l'utilisateur et le système d'exploitation, il interprète les commandes saisies par l'utilisateur et les traduit en instructions que le système d'exploitation.

Le terminal est l'application qui fournit l'environnement visuel pour interagir avec le shell

Nativement le shell d'Ubuntu est basé sur Bash, il est déjà utilisable en l'état mais nous allons le configurer pour plus de confort.

Pour cela nous installer zsh qui est considéré comme une amélioration de Bash.

Installer zsh.

```
sudo apt-get install zsh
```

Exécuter zsh pour la configuration.

```
zsh
```

Un prompt vous demande de choisir entre différentes options pour la configuration de zsh, sélectionner l'option qui configurera automatiquement le fichier `.zshrc` en tapant `2`.

Dès lors le shell est dorénavant basé sur zsh, cependant si vous ouvrez une autre fenêtre de terminal elle sera de nouveau basé sur bash. Pour résoudre cela nous allons configurer zsh comme shell par défaut.

```
chsh -s $(which zsh)
```

Il sera nécessaire de déconnecter puis reconnecter votre session pour que le changement soit actif.

## 2. Installer OhMyZsh

 [Documentation officielle OhMyZsh](#)

Pour nous faciliter la tâche dans la configuration et la personnalisation du terminal nous allons installer un framework zsh appelé OhMyZsh, cela aura pour effet de changer l'interface du terminal.

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

### 3. Installer PowerLevel10k

 [Dépôt Git officiel de PowerLevel10k](https://github.com/romkatv/powerlevel10k)

PowerLevel10k est un thème pour terminal zsh, sa configuration est simple et ajoutera la touche finale à notre terminal, son installation est encore plus simple grâce à OhMyZsh.

```
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-zsh/custom}/themes/powerleve
```

Pour rendre le thème actif il est nécessaire d'indiquer à zsh que nous souhaitons l'utiliser, pour cela il est nécessaire d'éditer le fichier `.zshrc` à la racine du dossier utilisateur.

```
sudo nano .zshrc
```

Remplacer le thème actuel par celui de PowerLevel10k.

```
...  
ZSH_THEME="powerlevel10k/powerlevel10k"  
...
```

Après avoir sauvegardé le fichier, un prompt de configuration de PowerLevel10k s'exécute, vous n'avez plus qu'à répondre aux questions en sélectionnant vos réponses grâce aux touches correspondantes indiquées.

Si le prompt n'apparaît pas, exécuter la commande `p10k configure`, vous pourrez recommencer la configuration à l'aide de cette même commande à n'importe quel moment.

#### 4. Installer l'auto suggestion

 [Dépôt officiel de zsh-autosuggestions](#)

Zsh-autosuggestion va apporter l'auto complétion des commandes du terminal, cela évite d'avoir à resaisir une commande utilisée couramment, voir retrouver des commandes complexes saisies précédemment et dont vous auriez oublié la syntaxe.

```
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-autosuggestions
```

Une fois le clonage du dépôt terminé, il est nécessaire d'ajouter "zsh-autosuggestions" à la liste des plugins dans votre fichier `.zshrc` à la racine du dossier utilisateur.

```
plugins=(
  ...
  zsh-autosuggestions
)
```

Lancer une nouvelle fenêtre de terminal, vous devriez avoir l'auto complétion d'activer.

## Configurer Git

 [Documentation officielle pour la configuration de Git](#)

### 1. Configurer votre utilisateur Git

Configurer le nom de votre utilisateur Git ( remplacer `<nom_utilisateur>` par le nom d'utilisateur souhaité ).

```
git config --global user.name "<nom_utilisateur>"
```

Configurer l'email de votre utilisateur Git ( remplacer `<email_utilisateur>` par l'email utilisateur souhaité ).

```
git config --global user.email "<email_utilisateur>"
```

( **optionnel** ) Configurer le nom de la branche par défaut lorsque vous initialisez un nouveau dépôt Git ( remplacer `<nom_branche>` par le nom de branche souhaité ).



Dans la majorité des cas on préférera utiliser “main” ou “master”.

```
git config --global init.defaultBranch <nom_branche>
```

Vous pouvez voir le nom et l'email utilisateur de votre configuration à l'aide des commandes suivantes :



```
git config user.name  
git config user.email
```

## 2. Lier une clé SSH à votre compte Github ou Gitlab



Il ne faut pas confondre Git avec Github ou Gitlab.

Git est le système de contrôle de versions, tandis que Github et Gitlab sont des plates-formes qui s'appuient sur Git pour fournir des fonctionnalités de collaboration et de gestion.



Une clé SSH ( Secure Shell Key ) est une paire de clé cryptographique ( un clé privée et une clé publique ) utilisée pour sécuriser les communications / connexions entre deux systèmes.

Pour pouvoir cloner des projets depuis Github ou Gitlab, ainsi que versionner nos projets il est nécessaire de lier une clé SSH au compte Github ou Gitlab.

Rendez-vous dans le dossier `.ssh` à la racine du dossier utilisateur.

```
cd ~/.ssh
```

Générer une clé ssh ( remplacer `<email_utilisateur>` par le même email que celui de votre utilisateur Git ).

```
ssh-keygen -o -t rsa -C "<email_utilisateur>"
```

Vous pouvez d'ores et déjà afficher votre clé publique dans le terminal car vous en aurez besoin dans l'étape suivante, les clés publiques ont l'extension `.pub`.

```
cat ~/.ssh/id_rsa.pub
```

Un prompt vous demande d'indiquer le chemin où sauvegarder la clé ou encore une "passphrase", vous pouvez appuyer sur `entrer` pour chacune des questions.

Rendez-vous dans sur Github ou Gitlab pour lier la clé générée et chercher le menu correspondant dans les paramètres de votre compte. Cliquer sur "Ajouter une clé SSH" et donner lui un label, puis coller votre clé publique.

Pour vérifier que la configuration de votre utilisateur Git est bonne vous pouvez cloner le projet suivant ( ou un autre, peu importe ).

```
git clone git@github.com:ijin/clone-test.git
```



Puisque nous avons lié une clé SSH de notre machine à notre compte Github ou Gitlab penser à sélectionner le lien "ssh" et non pas "https" lorsque vous clonez un projet.

## Configurer l'apparence de son système



Gnome est l'environnement de bureau installé par défaut sur Ubuntu.

Un moyen facile de customiser l'apparence de son environnement de bureau est d'utiliser les extensions Gnome disponibles sur le site [Extension Gnome](#).

Pour profiter de ces extensions il sera nécessaire d'installer une extension navigateur ainsi que deux paquets sur notre machine, ici nous utiliserons Google Chrome.

### 1. Installer l'extension pour Google Chrome

Installer l'extension Gnome shell pour Google Chrome [Chrome web store Gnome Shell](#).

### 2. Installer le paquet Chrome Gnome Shell

Installer le paquet nécessaire à l'interaction entre l'extension Google Chrome et la machine.

```
sudo apt install chrome-gnome-shell
```

Dès lors un bouton switch on / off apparaît sur les pages des extensions du site [Extension Gnome](#), vous pouvez dorénavant installer / désinstaller les extensions que vous souhaitez.

### 3. Installer le paquet Gnome Shell Extension Preferences

Cette application permet d'accéder aux paramètres de vos extensions.




```
sudo apt install gnome-shell-extension-prefs
```

#### 4. Installer l'application Gnome Tweaks ( optionnel )

Pour aller plus loin dans la configuration de l'interface il peut être intéressant d'installer l'application Gnome Tweaks, cette application permet la personnalisation de divers paramètres du système ( paramètres qui ne sont pas nativement présents dans l'application Settings fourni par Ubuntu ).

```
sudo apt install gnome-tweaks
```

#### Liste d'extensions intéressantes

-  [Dash to Panel](#) permet de utiliser une barre de tâches classique au lieu du Dash natif.
-  [User Themes](#) permet de installer et d'utiliser des thèmes pour Ubuntu.
-  [Gtile](#) permet de gérer l'affichage de ses fenêtres grâce à une grille.