

# Compte Rendu N°1

## Introduction

### Équipes

Cette SAE se fait par équipe donc nous avons constituer une **équipe de 3** :

- *Sebastian Novak*
  - [sebastian.novak.etu@univ-lille.fr](mailto:sebastian.novak.etu@univ-lille.fr)
  - Groupe D
- *GAVOILLE florian*
  - [florian.gavoille.etu@univ-lille.fr](mailto:florian.gavoille.etu@univ-lille.fr);
  - Groupe D
- *Cerdan sullivan*
  - [cerdan.sullivan.etu@univ-lille.fr](mailto:cerdan.sullivan.etu@univ-lille.fr)
  - Groupe D

### Semainier

Semaine	Étape	Suivi (1h)	Rendu
1-S06	Préparation VM par pré-configuration	X	
2-S07	Apprentissage balisage léger - base		
3-S08	vacance		
4-S09	Recherche / étude applications clientes	X	X
5-S10	Gitea : installation & utilisation du service		
6-S11	Apprentissage balisage léger - style + export	X	
7-S12	-		X

## Rapport SAE 2.03 - Semaine 1

Réponses aux questions et commandes effectuées

### Configuration matérielle dans VirtualBox

#### Question 1 :

Le **64 bit** dans **Debian 64 bit** indique que cette version de l'OS peut accéder à un total de  $2^{64}$  adresses mémoires, soit **16 milliards de gigaoctets** de mémoire. D'un point de vue matériel, les

systèmes *32 bit* peuvent avoir un maximum de 4 Go de RAM, alors que les systèmes *64 bit* n'ont pas réellement de limite<sup>[1]</sup>

[Source](#)



Les ordinateurs et OS récents fonctionnent sur une architecture **64 bits**

### Question 2 :

Le mode de connexion utilisé par défaut par *Virtualbox* est le **NAT**. Cette méthode permet de créer un réseau local virtuel propre à la VM et dont Virtualbox est le routeur. Autrement dit, sur le réseau local de la machine physique, la VM n'aura pas d'adresse IP attribuée, la seule adresse qui lui sera attribuée sera au sein du réseau virtuel créé.<sup>[2]</sup>

[Source](#)



Il existe d'autres manières de configuration réseau, pas par défaut et **plus complexes**, plus optimisées que le NAT

### Question 3 :

Le fichier de configuration s'appelle `sae203.vbox` et se trouve dans le dossier `/usr/local/virtual_machine/infoetu/login/vbox_vms/sae203`. Il est écrasé à chaque sauvegarde de la machine.

[Source](#)

### Question 4 :

Il **n'est pas recommandé** de modifier la configuration de la VM depuis le fichier **XML**. La documentation officielle préconise de modifier via l'interface graphique ou le terminal.

[Source](#)

## Installation OS de base

### Question 1 :

Un fichier **ISO**, du format ISO 9660 ou 13346, contient **toutes les données que l'on souhaite transférer sans les compresser** (contrairement à un fichier d'archive). Il contient fréquemment **toutes les données** et la **structure** nécessaires pour lancer un système d'exploitation ou un programme depuis un support externe (CD, DVD, Clé USB).

[Source](#)

### Question 2 :

**MATE** et **GNOME** sont des **environnements graphiques** utilisés principalement sur **Debian**. Ils définissent comment les informations seront **affichés** à l'écran et le **style** de ces derniers. Autrement dit, l'environnement graphique définit le comportement de **l'interface graphique**.

[Source](#)

### Question 3 :

Un serveur Web a différentes définitions, l'une d'un point de vue **logiciel**, l'autre **matériel**.

### **Software (logiciel)**

Un serveur Web d'un point de vue *logiciel* contient différents **fragments** qui contrôlent la façon dont l'utilisateur peut accéder aux fichiers hébergés. Il comprend au minimum un **serveur**

**HTTP** qui contient les url et le protocole HTTP.

## Hardware (matériel)

Un serveur Web d'un point de vue *matériel* est un ordinateur qui stocke **tous les fichiers d'un site web** et qui les **envoie** aux appareils qui se rendent sur le site.

Il y a deux types de serveurs web :

- |                  |   |
|------------------|---|
| <b>Statique</b>  | Composé d'un <b>ordinateur</b> et d'un <b>serveur HTTP</b> , il envoie les fichiers hébergés <b>tels quels</b> au navigateur.   |
| <b>Dynamique</b> | Il contient, en plus du serveur statique, des <b>composants logiciels</b> , comme des <i>serveurs d'applications</i> ou des <i>bases de données</i> . Il <b>met à jour les données</b> avant de les envoyer au serveur via HTTP. <a href="#">Source</a> |

### Question 4 :

Un serveur SSH établit un lien **sécurisé** entre un utilisateur et un serveur. Il permet **d'accéder** à des fichiers sur un serveur, qui seront **chiffrés** durant le transfert, et **d'effectuer des commandes** à distance.

[Source](#)

### Question 5 :

Un *serveur mandataire* (ou *proxy*) est un programme **intermédiaire** utilisé lors de la navigation sur internet. En plus de **faciliter** l'accès au **web**, il peut **recevoir** les demandes du client et **retourner** les réponses. Dans certains cas, il peut aussi **transmettre** des requêtes. Il existe deux types de proxy :

- |                      |  |
|----------------------|--|
| <b>Proxy direct</b>  | Il gère les demandes <b>depuis et vers n'importe où sur internet</b> .                                       |
| <b>Proxy inverse</b> | Il prend des <b>requêtes</b> d'internet pour les <b>envoyer</b> vers un serveur d'un <i>réseau interne</i> . |

[Source](#)

## Sudo

Commandes utilisées :

⇒ Ajouter l'utilisateur **user** au groupe des sudoers

```
sudo usermod -aG sudo user
```

⇒ Afficher les groupes du **user**

```
groups user
```

# Suppléments invité

## Question 1 :

Pour récupérer la version de notre noyau, nous pouvons effectuer la commande suivante dans un terminal :

```
uname -r
```

On remarque que notre noyau est sur la version 6.1.0-30-amd64.

## Question 2 :

Les compléments invités permettent d'améliorer l'utilisation de virtualbox avec :

Ajout	Description
Meilleure intégration de la souris	Le pointeur peut passer plus facilement et fluidement de l'ordinateur à la VM
Presse-papier partagé	Tout ce que l'on copie de l'ordinateur sera aussi dans le presse-papier de la VM et inversement
Dossiers partagés	Accès plus facile aux dossiers de l'ordinateur depuis la VM
Meilleur support graphique	C'est grâce à cette fonctionnalité que l'on peut modifier la taille de la fenêtre de notre VM et que cette dernière s'adapte totalement

Cette liste est non exhaustive mais constitue les principaux changements.

[Source](#)

## Question 3 :

La commande **mount** permet d'**attacher** une partition **externe** au système (pouvant être un disque externe comme une autre partition du disque principal) au **système de fichier /**.

Un **exemple** simple est lorsque l'on sépare le **/home** de la partition **/**, on crée une partition externe qui sera **montée** à chaque lancement et rattachée au fichier **/home**.

Dans notre cas, on souhaite accéder aux fichiers contenus dans le CD-ROM. Pour cela, on va définir le dossier **/mnt** comme point de montage, nous permettant d'accéder aux fichiers du CD-ROM par **/mnt**.

Source : **man mount**

# Quelques questions sur la documentation Debian

## Question 1 :

Debian est une distribution **GNU Linux** qui se veut *libre* et de *qualité supérieure*. Développé entièrement **bénévolement**, la distribution Debian est **stable**, **complète** <sup>[3]</sup>, **gratuite** tant à l'utilisation qu'à l'amélioration et la redistribution et **active** grâce au travail régulier de plus de

**1600 bénévoles.**

Le nom Debian vient de la contraction de **Debra** et **Ian** Murdock, les deux créateurs originels du projet.

[Source](#)

### Question 2 :

Il y a **3 équipes** support différentes pour les différentes durées de prise en charge :

#### **Debian stable support**

L'équipe chargée du **support** des **dernières version stables** de Debian. Cette équipe est composée des **équipes de sécurité et de Release de Debian** et s'occupent de mettre à jour la dernière version pour la **maintenir stable**. Durée : *environ 3 ans*

#### **LTS / Debian oldstable support**

Cette équipe est chargée de **prolonger** le support des **anciennes versions** pour que ces dernières soient stables pendant 5 ans. L'équipe travaille sur les versions qui **ne sont plus assurées** par le **Debian stable support** et est composée de **bénévoles indépendants** des équipes release et sécurité de Debian. Durée : *environ 2 ans*

#### **ELTS support**

La **dernière étape** avant qu'une version **ne reçoive plus d'aide support**. Cette équipe, agissant **après** le LTS support, reste tout autant **bénévole et indépendante** des équipes release et sécurité Debian. Elle agit pendant **5 ans**<sup>[4]</sup> pour qu'une version puisse atteindre les **10 ans de vie**.

[Source LTS et Debian Stable](#)

[Source ELTS](#)

[Source Releases](#)

### Question 3 :

La durée de vie d'une version dépend de l'arrivée des **équipes support, LTS et ELTS**. En effet, les versions **avant la 2.0** n'ont pas reçu de support. **Les versions avant Debian 6** n'ont reçu que du support pendant **3 ans ou moins**, assuré par les **équipes release et sécurité** Debian. La **version 2.1** est une exception car elle a reçu du support LTS pendant... *1 mois seulement*. L'équipe **LTS** n'a ensuite plus été sollicité jusqu'à la version **Debian 6**, qui a été **la première** à recevoir **2 ans de support LTS**, amenant sa durée de vie à **5 ans**. **Debian 7** a ensuite inauguré **l'ELTS** avec **2 ans de support supplémentaires**, amenant sa durée de vie à **7 ans**. Les **version ultérieures** ont reçu et recevront les durées de support citées précédemment pour atteindre **10 ans** de durée de vie.

*Les sources sont les mêmes que pour la question 2*

### Question 4 :

Il y a au minimum **1 version** activement maintenue par Debian. [Source](#)

### Question 5 :

Les noms de code sont basés sur les **noms des personnages** de **Toy Story**. En effet, la personne ayant repris le lead du projet Debian après Ian Murdock, **Bruce Perens**, travaillait à **Pixar**, le studio à l'origine de la saga Toy Story. C'est ainsi qu'on retrouve tous les personnages de la série dans les noms de version Debian.

[Source](#)

**Question 6 :**

Il y a **9 architectures** prises en compte :

- amd64
- i386
- ppc64el
- s390x
- armel
- armhf
- arm64
- mipsel
- mips64el

[Source](#)

**Question 7 :**

La première version avec un nom de code était **Debian 1.1**, sous le nom de **Buzz** pour Buzz l'éclair, a été annoncée le **17 juin 1996**. Elle n'aura duré que quelques mois puisque la version Rex la remplacera en décembre de la même année.

[Source](#)

**Question 8 :**

La dernière version annoncée à ce jour est **Debian 15** sous le nom de **Duke**, qui a été annoncée le **22 janvier 2025**.

[Source](#)

## ***2. Tutoriel pour l'installation d'une VM automatisée :***

Afin de mener à bout ce processus vous aurez besoin de :  
L'application Oracle Virtual Box  
Les fichiers de configuration et VISO de Moodle mentionnées sur le PDF de la SAE.

L'installation va s'effectuer en 7-8 étapes :

- ☐ A l'aide de Virtual Box, créer une VM avec ces spécifications :
  - Type : Linux
  - Version : Debian 64-bit
  - Mémoire vive (RAM) : 2048 Mo pour être à l'aise à l'usage.

- Disque dur : 20 Go, ne pas cocher la case "Pre-allocate Full Size"
  - Cocher la case "Skip Unattended Installation" pour éviter que Virtualbox réalise des actions non souhaités.
- Extraire depuis Moodle le fichier zip avec le Viso et les fichiers de configuration pour l'installation et placer les contenus dans le dossier où vous avez fait la VM
- Remplacer la chaîne @@UUID@@ du fichier S203-Debian12.viso par un identifiant unique universel. Le plus simple est d'exécuter la commande ci-dessous en étant placé dans le même répertoire que votre fichier : `sed -i -E "s/(--iprt-iso-maker-file-marker-bourne-sh).*$/\1=$(cat /proc/sys/kernel/random/uuid)/" S203-Debian12.viso`
- Dans le fichier **preseed.cfg** qui se trouve au même endroit que le **viso** on doit ajouter quelques commandes pour automatiser totalement l'installation. Ces commandes vont installer quelques applications dont nous avons besoin comme **git** ou **bash-completion** et ajouter l'utilisateur normal au groupe **sudo**.
- A la ligne 83 : `tasksel tasksel/first multiselect standard ssh-server mate-desktop`
  - A la ligne 84 : `d-i pkgssel/include string git sudo sqlite3 curl bash-completion neofetch`
  - A la ligne 56 : `d-i passwd/user-default-groups string audio cdrom video sudo`



Fun Fact : La recherche de ces commandes nous a mené sur un forum Google

Le fichier ressemblera à ceci :

```
78
79  ## Installation meta-paquetages
80  # Tâches à installer (via des méta-paquetages)
81  #   Lister les possibilités : tasksel --list-task (en ligne de commande)
82  #   Utiliser au minimum "standard" est une bonne idée
83  tasksel tasksel/first multiselect standard ssh-server mate-desktop
84  d-i pkgssel/include string git sudo sqlite3 curl bash-completion neofetch
```

Fig1 : Ajout des applis.

```
45  ### Ajout des comptes root et user
46  d-i user-setup/allow-password-weak boolean true
47  ## root
48  d-i passwd/root-login boolean true
49  d-i passwd/root-password password root
50  d-i passwd/root-password-again password root
51  ## Utilisateur standard
52  d-i passwd/user-fullname string User
53  d-i passwd/username string user
54  d-i passwd/user-password password user
55  d-i passwd/user-password-again password user
56  d-i passwd/user-default-groups string audio cdrom video sudo
```

Fig2 : Ajout de l'utilisateur au groupe audio, cdrom, video et sudo.

- Aller dans la configuration de la VM et mettre comme **ISO** d'installation le fichier **viso** auquel on a appliqué la commande **sed** précédemment.

- ☐ Demarrer la VM et attendre que l'installation se complete
- ☒ Et voila ! Vous avez une VM installee automatiquement

## Semaine 2 :

Un des **objectifs** de la sae 2.03 est de nous **apprendre un langage de balisage** utile pour la création de documentation mais pas que.

Pour montrer ce que nous avons appris, nous avons fait ce rapport entièrement en **asciidoc**, pour le moment il contient :

1. du texte en gras
2. de l'insertion de code source
3. de l'insertion de lien cliquable
4. des images avec modification de leur taille initiale
5. un tableau complexe ( le semainier )
6. des listes
  - a. Ordonnée
  - b. non ordonnée
7. un sommaire automatique

Donc pour le rapport final, nous devons encore mettre en place :

- le css
  - mettre une page de garde
  - changer la police d'écriture si possible
    - couleurs
    - font-family
- la gestion de multifichier
- refaire un tableau complexe
- refaire la mise en page des rapports à integrer dans le rapport final avec les connaissances apprises en plus.

Nous avons de choix à notre disposition **2 formats** au choix:



- Le format [Markdown](#) dans sa version étendue par le logiciel [pandoc](#)
- Le format [AsciiDoctor](#) (version étendue du format AsciiDoc)

Nous avons fini par choisir le format **AsciiDoctor** car selon notre professeur il est plus complet et demande moins d'extension que le **Markdown**.

## Installation de Asciidoctor

Avant d'installer **Asciidoctor** nous avons du installer le **RubyInstaller**. Pour cela nous sommes simplement aller sur le site de [RubyInstaller](#) et avons cliquer sur installer, cela ne fonctionne que pour **Windows**. Pour **Debian** il suffit de taper la commande : `sudo apt-get install ruby-full`

Ensuite une fois ruby installé nous avons dû exécuter une commande dans le **RubyInstaller**. `gem install asciidoctor`

Afin de vérifier la réussite de l'installation il a fallu faire : `asciidoctor --version`

## Installation de asciidoctor-pdf

Pour installer **asciidoctor-pdf** nous avons dû exécuter une commande dans le **RubyInstaller**. `gem install asciidoctor-pdf`

Asciidoctor-PDF est une extension d'Asciidoctor qui permet de générer des documents au format PDF à partir de contenu rédigé en AsciiDoc.

## Commandes utile

Commande pour convertir un fichier en asciidoctor en pdf :

- Il faut se mettre dans le dossier qui contient le fichier **.adoc** et écrire :
  - directement dans son terminal si on est sous **Linux**
  - dans le terminal en passant par **RubyInstaller** si on est sous **Windows**

et taper `asciidoctor-pdf <nom_du_fichier>`

Commande pour convertir un fichier en asciidoctor en html :

- Il faut se mettre dans le dossier qui contient le fichier **.adoc** et écrire :
  - directement dans son terminal si on est sous **Linux**
  - dans le terminal en passant par **RubyInstaller** si on est sous **Windows**

et taper `asciidoctor <nom_du_fichier>`

## Rapport SAE 2.03 - Semaine 3

# 1. Configuration de Git

Ouvrez un terminal et effectuez les commandes suivantes :

```
git config --global user.name "Prénom Nom" ①  
git config --global user.email "votre@email" ②  
git config --global init.defaultBranch "master" ③
```

Les deux premières commandes permettent de s'identifier à notre compte git. Le paramètre `--global` permet de sauvegarder cette configuration sans devoir la remettre à chaque connexion.

La troisième commande permet de créer automatiquement une branche master lors de l'exécution de la commande `git init` (donc à la création d'un dépôt).

## 2. Interfaces graphiques

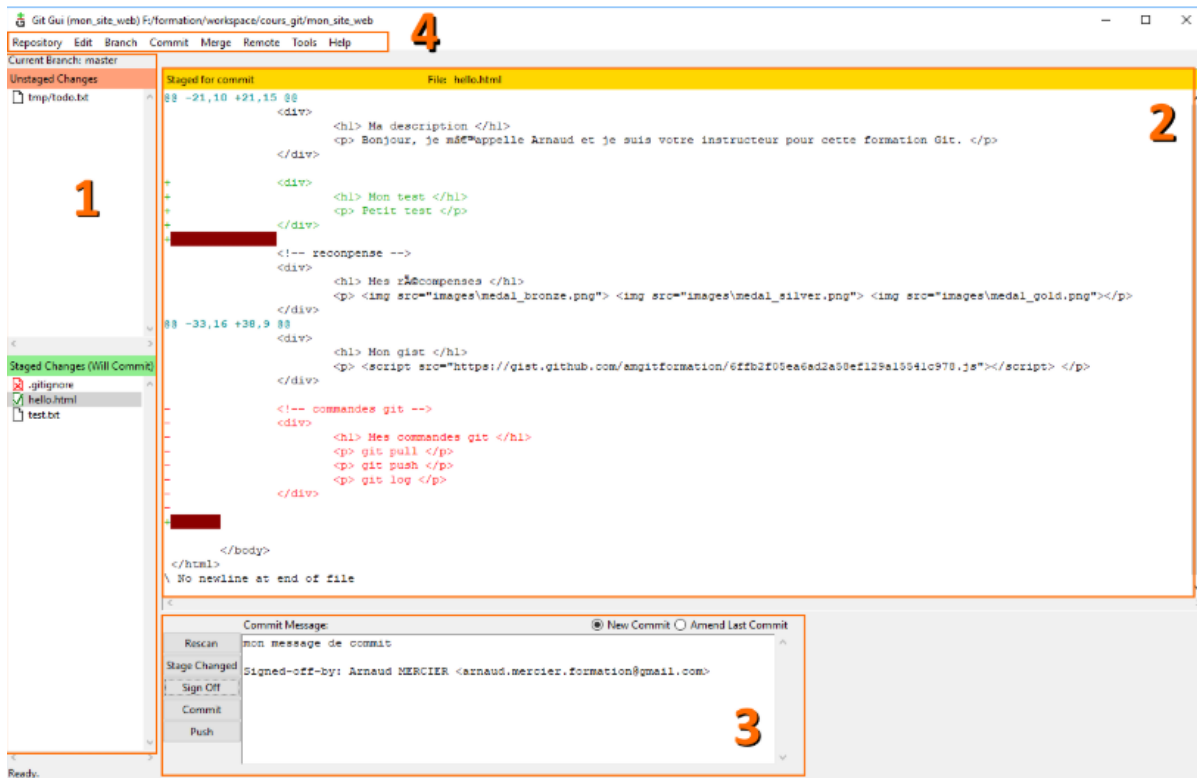
- Qu'est-ce que le logiciel gitk ? Comment se lance-t-il ?
- Qu'est-ce que le logiciel git-gui ? Comment se lance-t-il ?

Les deux logiciels sont des interfaces graphiques pour la commande git. Gtik permet de voir l'historique des depots graphiquement, tandis que git-gui est une interface graphique complete, permettant la gestion des commits.

On les lance avec :

```
gitk  
  
git gui
```

Voici l'interface de git-gui avec une explication brève de l'interface :

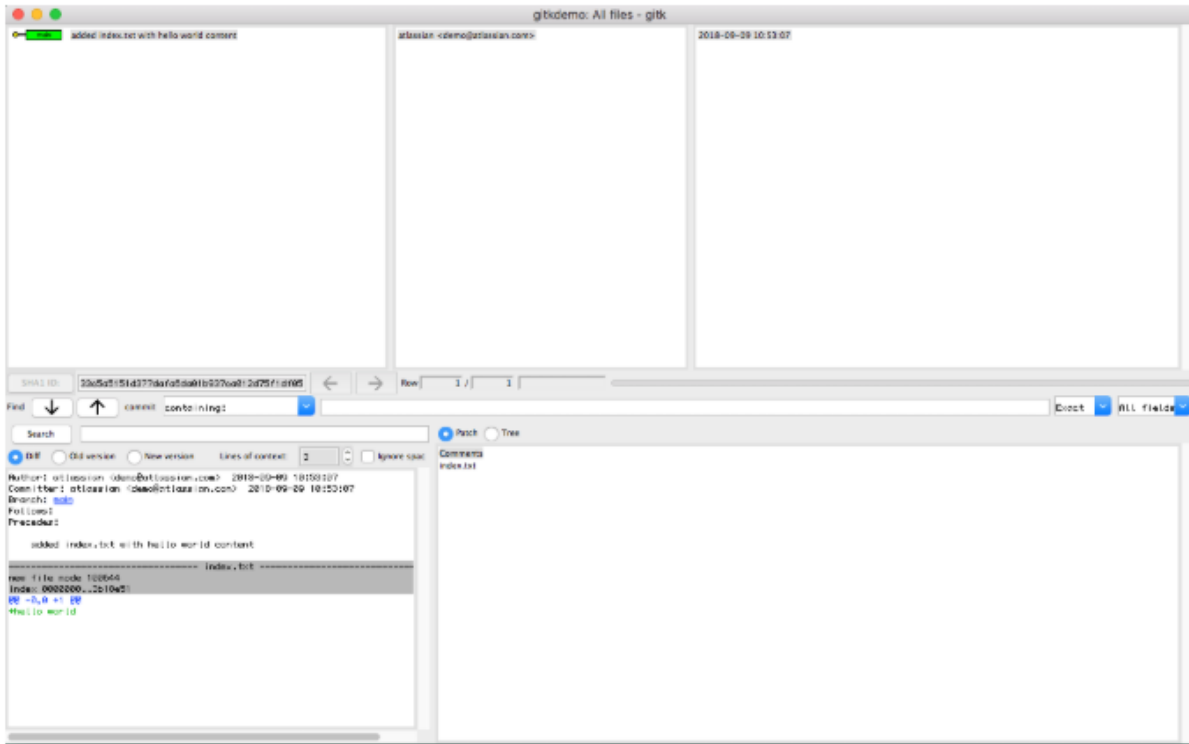


Git Gui

Voici la vue d'ensemble de l'interface graphique Git Gui. Nous allons voir chacun de ces points dans la suite de l'article:

1. État du workspace en haut et état de la zone d'index en bas. De plus, tout en haut nous pouvons trouver le nom de la branche courante. C'est finalement l'équivalent de la commande: **git status**.
2. Modifications courantes du fichier sélectionné dans la partie (1). C'est l'équivalent de la commande: **git diff [mon\_fichier]**.
3. Permet en outre de réaliser ou modifier des commits puis de les pousser sur le remote.
4. Barre de menu de l'outil qui regroupe l'ensemble des actions disponibles dans les autres parties de l'interface et bien plus encore.

De même pour l'interface de gitk :



Le volet supérieur gauche affiche les commits du dépôt, avec le plus récent en haut. Le volet inférieur droit affiche la liste des fichiers impactés par le commit sélectionné. Le panneau inférieur gauche affiche les informations du commit et le diff complet. En cliquant sur un fichier dans le volet inférieur droit, le diff dans le volet inférieur gauche se concentre sur la section concernée.

### 3. Autres interfaces

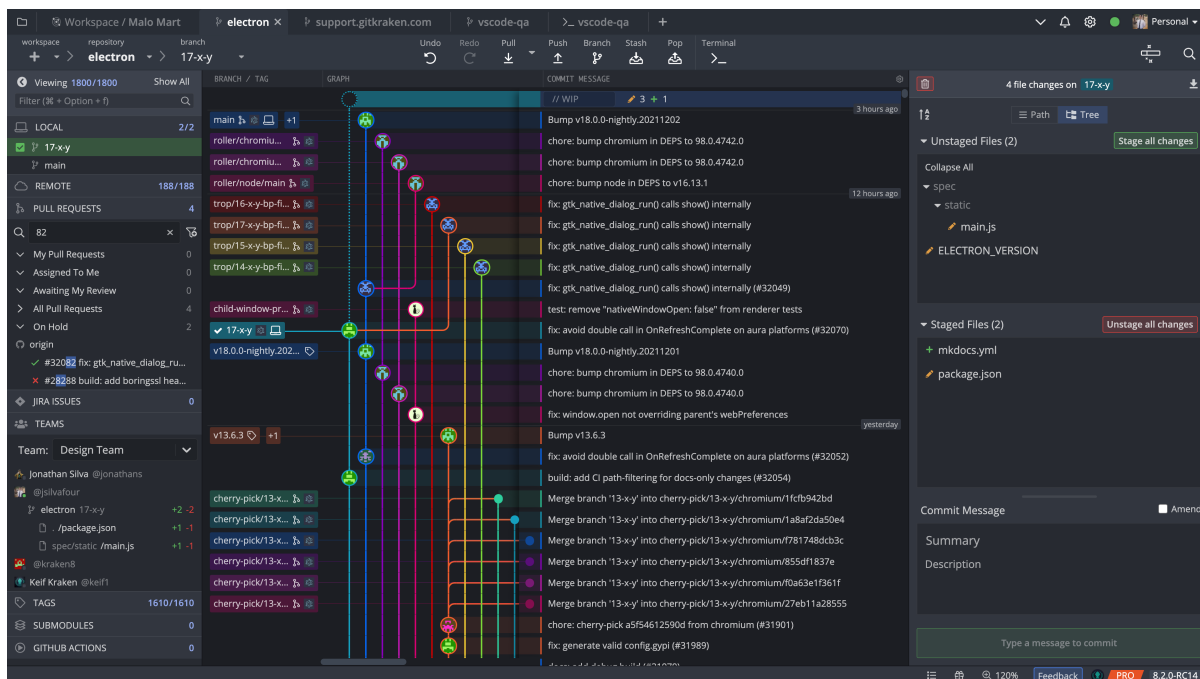
Git dispose d'un **grand** nombre d'interfaces graphiques utiles, l'un d'entre eux étant [GitKraken](#), une interface graphique très populaire. Sa popularité est une des raisons pour laquelle on l'a choisi mais aussi pour son design ressemblant à Visual Studio Code qui nous a beaucoup plu.

Pour l'installer on a eu besoin de deux commandes :

```
wget https://release.gitkraken.com/linux/gitkraken-amd64.deb

sudo apt install ./gitkraken-amd64.deb
```

Voici une image de l'interface de gitkraken :



## Comparaison a gitk et git-gui :

Avantages	Desavantages
Une interface facile a comprendre et plus riche comparee a gitk et git-gui	Peut etre complique a prendre en main pour les debutants
Les fonctionnalites de gitk et git-gui sont presentes dans gitkraken et on retrouve encore plus de fonctionnalites utiles telles que le Workspace Management qui permet d'organiser son espace de travail de maniere efficace	C'est une interface graphique connue pour etre assez lente
Une appli facilement personnalisable pour l'ajuster a ses preferences	Existence d'un centre d'aide avec des problemes frequemment rencontres mais aucun forum officiel permettant aux utilisateurs de faire part de problemes plus particuliers

[1] Il est impossible d'avoir 16 milliards de giga octet de RAM à l'heure actuelle

[2] Cette méthode peut être contournée en ajoutant une règle de redirection de port dans la configuration de la machine virtuelle

[3] La documentation parle de presque 65 000 paquets de logiciels disponibles à l'installation

[4] Sauf la version Debian 7 "Wheezy" qui elle n'a reçu que 2 ans de support ETLs, et les versions antérieures n'ayant pas reçu de support ELTS