# Compte Rendu final

# Table of Contents


Introduction . . . 2
- Équipes . . . 2
- Semainier . . . 2

Rapport SAE 2.03 - Semaine 1 . . . 2
- Configuration matérielle dans VirtualBox . . . 2
- Installation OS de base . . . 3
- Sudo . . . 4
- Suppléments invité . . . 4

Quelques questions sur la documentation Debian . . . 5
- *Tutoriel pour l'installation d'une VM automatisée :* . . . 7

Semaine 2 : . . . 9
- Installation de Asccidoctor . . . 10
- Installation de asciidoctor-pdf . . . 10
- Commandes utile . . . 10

Semaine 4 : . . . 10

Rapport SAE 2.03 - Semaine 3 . . . 11
- 1.Configuration de Git . . . 11
- 2.Interfaces graphiques . . . 11
- 3.Autres interfaces . . . 13
- Comparaison a gitk et git-gui : . . . 14

Rapport SAE 2.03 - Semaine 4 . . . 14
- Redirection de port . . . 14
- Reponses aux questions : . . . 15

Installation de Gitea . . . 15
- Télécharger le fichier . . . 15
- Vérifier les signatures GPG . . . 16
- Configurer un utilisateur gestionnaire . . . 16
- Préparation de l'environnement . . . 16
- Finalisation . . . 16
- Premier lancement . . . 17
- Paramètrage au premier lancement . . . 17

# Introduction

## Équipes

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

- *Sebastian Novak*
  - sebastian.novak.etu@univ-lille.fr
  - Groupe D
- *GAVOILLE florian*
  - florian.gavoille.etu@univ-lille.fr;
  - Groupe D
- *Cerdan sulivan*
  - cerdan.sulivan.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 : Que signifie "64-bit" dans "Debian 64-bit" ?**
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[1]
Source : Microsoft, différences entre 32 bits et 64 bits

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

**Question 2 : Quelle est la configuration réseau utilisée par défaut ?**
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.[2]
Source : IT Connect, comprendre les différents types de réseaux VirtualBox

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

**Question 3 : Quel est le nom du fichier XML contenant la configuration de votre machine ?**
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 : Documentation Virtualbox

**Question 4 : Sauriez-vous modifier directement ce fichier de configuration pour mettre 2 processeurs à votre machine ?**
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 : Documentation Virtualbox

# Installation OS de base

**Question 1 : Qu'est-ce qu'un fichier iso bootable ?**
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 : Ionos, qu'est-ce qu'un fichier iso ?

**Question 2 : Qu'est-ce que MATE ? GNOME ?**
**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 : Wikipedia, GNOME

**Question 3 : Qu'esst-ce qu'un serveur WEB ?**
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 :

| | |
|---|---|
| **Statique** | Composé d'un **ordinateur** et d'un **serveur HTTP**, il envoie les fichiers hébergés **tels quels** au navigateur. |
| **Dynamique** | Il contient, en plus du serveur statique, des **composants logiciels**, comme des *serveurs d'applications* ou des *bases de données*. Il **met à jour les données** avant de les envoyer au serveur via HTTP. Source : Mozilla Developer Network |

**Question 4 : Qu'est-ce qu'un serveur SSH ?**

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 : Hostinger, tutoriel SSH Linux

**Question 5 : Qu'est-ce qu'un serveur mandataire ?**

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 :

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

Source : Mozilla Developer Network

# 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 : Quel est la version du noyau de Linux utilisé par votre VM ?**

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 : A quoi servent les suppléments invités ?**
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 : site officiel de Virtualbox

**Question 3 : A quoi sert la commande mount ?**
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 : Qu'est-ce que le projet Debian ? D'où vient le nom ?**
Debian est une distribution **GNU Linux** qui se veut *libre* et de *qualité supérieure*. Développé entièrement **bénévolement**, la distribution Debian est **stable**, **complète** [3], **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 **Deb**ra et **Ian** Murdock, les deux créateurs originels du projet.

Source : Documentation debian, FAQ

**Question 2 : Il existe 3 durées de prises en charge de ces versions : la durée minimale, la durée en support Long terme et la durée en support long terme étendue. Quelles sont les durées de ces prises en charge ?**

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 recoive 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**[4] pour qu'une version puisse atteindre les **10 ans de vie**.

Source LTS et Debian Stable
Source ELTS
Source Releases

**Question 3 : Pendant combien de temps les mises à jour de sécurité sont-elles fournies ?**

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 excpetion 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 : Combien de version au minimum sont activement maintenues par Debian ?**

Il y a au minimum **1 version** activement maintenue par Debian. Source : Documentation Debian, page des releases

**Question 5 : Chaque distribution majeur possède un nom de code différent. Par exemple, la version majeur actuelle se nomme bookworm. D'où viennent les noms de code données aux distributions ?**

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 : Documentation de la version Buzz

**Question 6 : L'un des atouts de Debian fut le nombre d'architectures officiellement prises en charge. Combien et lesquelles sont prises en charge par la version Bullseye ?**

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

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

Source : Documentation Debian, page de la release Bullseye

**Question 7 : Informations sur la première version avec un nom de code :**
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 : Documentation Debian, page de la release Buzz

**Question 8 : Informations concernant la dernière version de Debian annoncée**
La dernière version annoncée à ce jour est **Debian 15** sous le nom de **Duke**, qui a été annoncée le **22 janvier 2025**.

Source : Documentation Debian, page de la release Duke

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

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

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

- ☐ A l'aide de Virtual Box, creer une VM avec ces specifications :
  - 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 ou 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 meme 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 pkgsel/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 mene sur un forum Google

Le fichier ressemblera a 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 pkgsel/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'user 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 applique la commande **sed** precedemment.
- [ ] Demarrer la VM et attendre que l'installation se complete
- [x] 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 crétion de documentation mais pas que.

Pour montrer ce que nous avons appris, nous avons fait ce rapport entièrement en asciidoctor, 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 avions 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 Asccidoctor

Avant d'installer **Asccidoctor** 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 éxecuter 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 éxecuter 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>`

# Semaine 4 :

Vacances

# 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'execution 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 ?

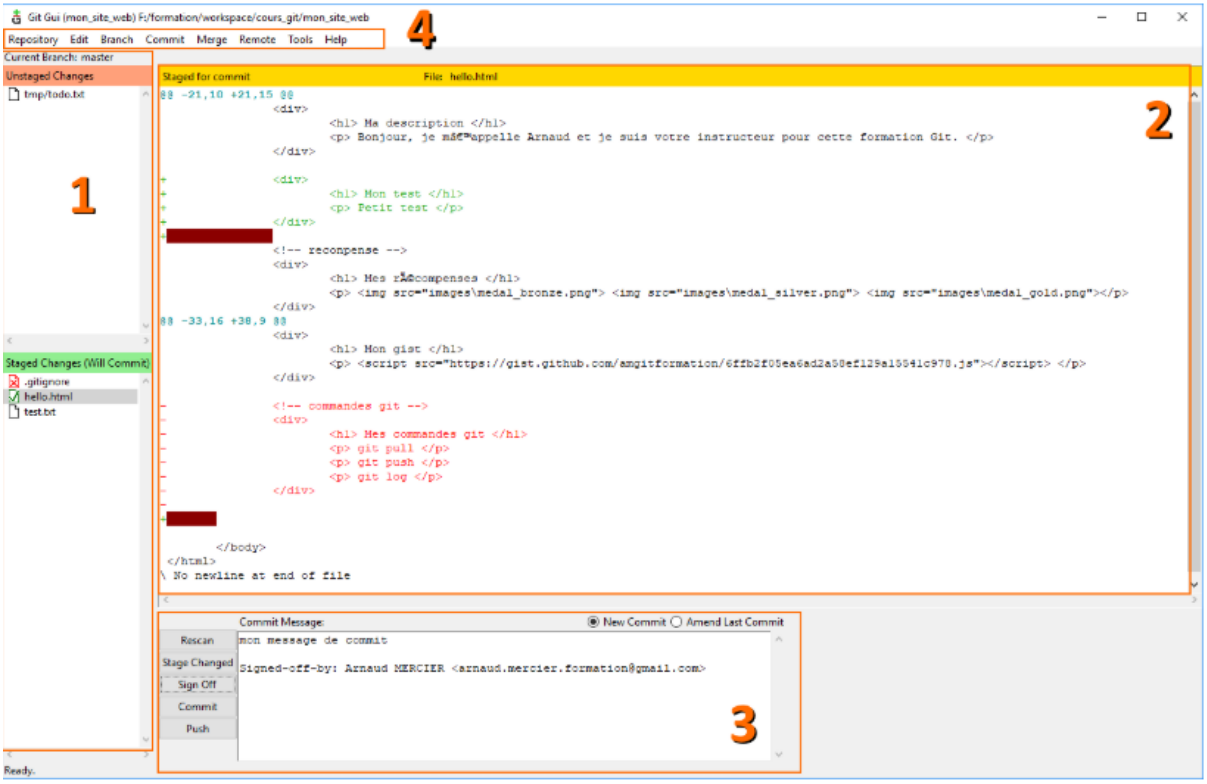
Les deux logiciels sont des interfaces graphiques pour la commande git. Gtik permet de voir l'historique des depots graphiquement, tandis que git-gui est une interface graphique complete, permettant la gestion des commits.

On les lance avec :

```
gitk

git gui
```

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

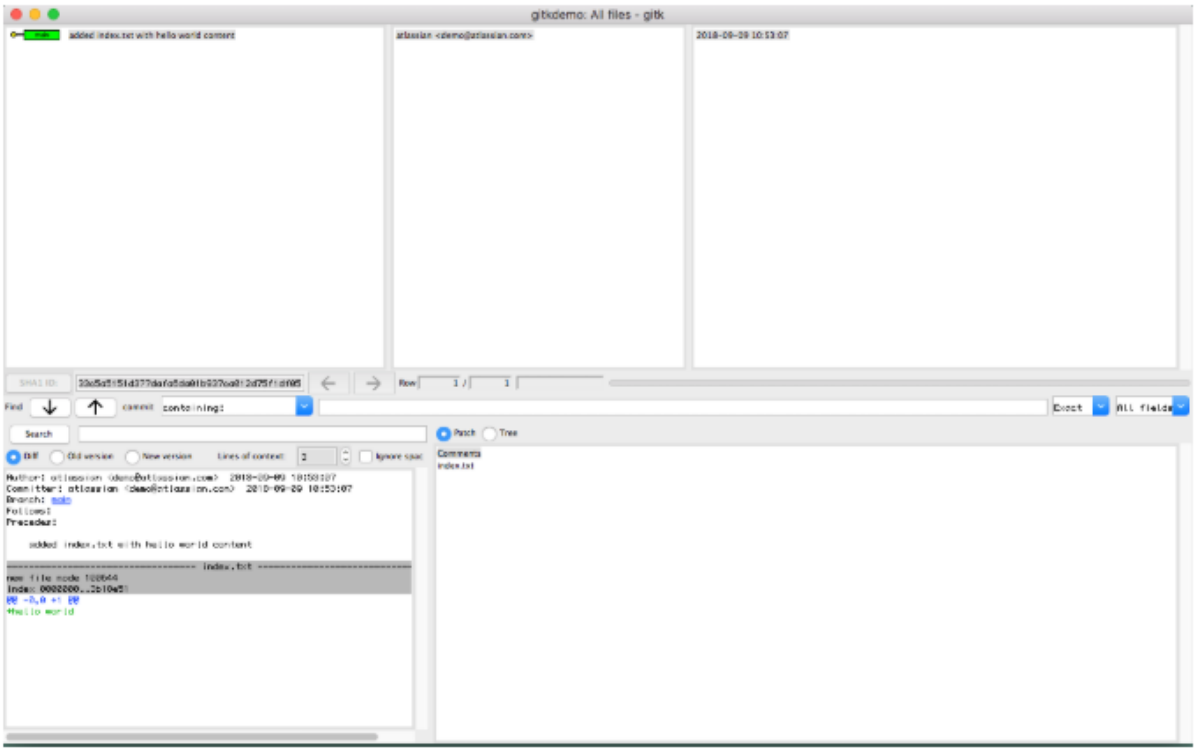



Git Gui

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

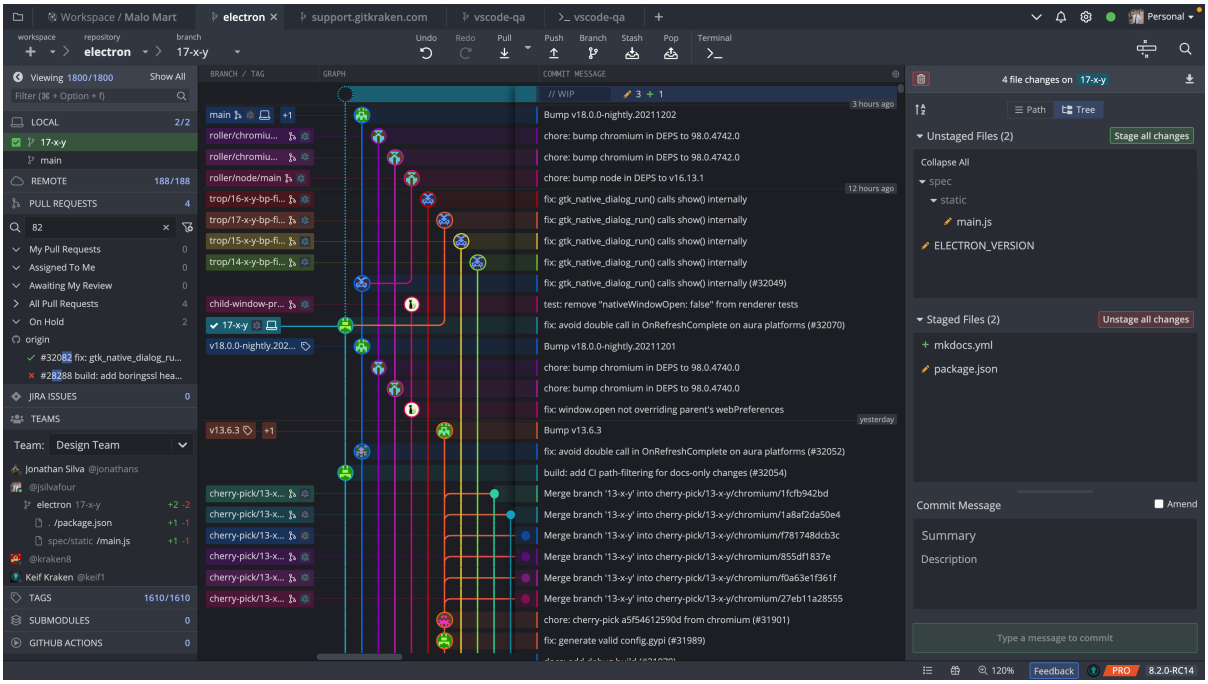
Git dispose d'un **grand** nombre d'interfaces graphiques utiles, l'un d'entre eux etant GitKraken, une interface graphique tres populaire. Sa popularite est une des raisons pour laquelle on l'a choisi mais aussi pour son design ressemblant a Visual Studio Code qui nous a beaucoup plu.

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 personalisable pour l'ajuster a ses preferences | Existence d'un centre d'aide avec des problemes frequemment rencontres mais aucun forum officel permettant aux utilisateurs de faire part de problemes plus particuliers |

# Rapport SAE 2.03 - Semaine 4

## Redirection de port

Pour redirectionner les ports sur Virtual Box la demarche est relativement simple :

**Etape 1 : Allez sur le menu de virtual box et appuyez sur le bouton "Configuration" :**

[500] | *../img/Redirection1.png*

**Fig 1 : Acceder a la configuration de la machine**

Etape 2 : Dans le menu de configuration, allez dans la categorie Reseau et appuyez sur le bouton

Avance pour ensuite appuyer sur le bouton Redirection de ports :

[500] | *../img/Redirection2.png*

**Fig 2 : Acceder a la rubrique reseau**

Etape 3 : Dans le menu de redirection des ports, appuyez sur le bouton d'ajout (entoure en rouge) et rentrer les memes elements que dans la **Fig 3** :

[500] | *../img/Redirection3.png*

**Fig 3 : Menu de redirection des ports**

# Reponses aux questions :

- Qu'est-ce que Gitea ?
- À quels logiciels bien connus dans ce domaine peut-on le comparer ?
- Qu'est-ce qu'un fork (dans le domaine du développement logiciel bien entendu) ?
- De quel logiciel Gitea est-il le fork ? Ce logiciel existe-t-il encore ?

Gitea est un logiciel "All-in-one" d'hebergement de developpement sous licence MIT. Similaire a Github et Gitlab, il permet d'heberger du code, de gestioner ses projets informatiques, ainsi que leur code. Le deploiement et la maintenance du code est aussi facile a effectuer ce qui fait de Gitea un bon outil pour les petites equipes ou developpeurs individuels.

Un fork est un nouveau logiciel cree a partir du code source d'un logiciel existant. C'est comme creer une nouvelle maison a partir d'un modele existant. Dans le cas de Gitea, il a ete fork a partir du logiciel Gogs et selon les developpeurs presque tout le code a ete change. Le logiciel Gogs existe encore.

Sources : Documentation de Gitea et Wikipedia : Fork

# Installation de Gitea

Cette partie a été rédigée à l'aide de la documentation officielle de Gitea, disponible via ce lien.

## Télécharger le fichier

La première étape est de télécharger la version de Gitea que vous souhaitez sur ce site. Il est recommandé pour Linux de choisir la version `linux amd64` (nous avons procédé ainsi). Il est également possible de passer par wget :

```
wget -O gitea https://dl.gitea.com/gitea/main-nightly/gitea-main-nightly-linux-amd64
chmod +x gitea
```

# Vérifier les signatures GPG

Gitea est sécurisé par des clés GPG afin d'éviter les modifications involontaires ou fatales du fichier binaire. Pour décrypter le fichier, voici les commandes à effectuer :

```
gpg --keyserver keys.openpgp.org --recv 7C9E68152594688862D62AF62D9AE806EC1592E2
gpg --verify gitea-main-nightly-linux-amd64.asc gitea-main-nightly-linux-amd64
```

# Configurer un utilisateur gestionnaire

Pour faciliter la gestion, nous allons créer un utilisateur `git` qui aura toutes les permissions (et qui sera le propriétaire) des fichiers gitea que l'on configurera.

La commande suivante permet de créer cet utilisateur :

```
adduser \
   --system \
   --shell /bin/bash \
   --gecos 'Git Version Control' \
   --group \
   --disabled-password \
   --home /home/git \
   git
```

# Préparation de l'environnement

Certains dossiers, qui nous seront utiles plus tard, sont à créer et configurer.

Ces commandes permettent de créer les dossiers obligatoires au bon fonctionnement de Gitea et à configurer les permissions pour l'utilisateur Git :

```
mkdir -p /var/lib/gitea/{custom,data,log}
chown -R git:git /var/lib/gitea/
chmod -R 750 /var/lib/gitea/
mkdir /etc/gitea
chown root:git /etc/gitea
chmod 770 /etc/gitea
```

# Finalisation

Ces commandes permettent de finaliser la configuration de Gitea :

```
export GITEA_WORK_DIR=/var/lib/gitea/
cp gitea /usr/local/bin/gitea
```

# Premier lancement

Pour lancer Gitea, exécutez la commande :

```
GITEA_WORK_DIR=/var/lib/gitea/ /usr/local/bin/gitea web -c /etc/gitea/app.ini
```

Cette commande lancera le serveur Gitea.

# Paramètrage au premier lancement

Après avoir lancé votre serveur, rendez-vous dans votre navigateur et allez sur l'adresse `localhost`

Nous configurerons le type de Base de données sur "sqlite3", puis créerons le compte admin de nom "gitea", de mot de passe "gitea" et d'adresse gitea@localhost

[500] | *../img/image_localhost_gitea.png*

Votre Gitea est désormais installé et configuré ! Vous pouvez l'utiliser dès à présent !

**Question : Comment voir la version actuelle de Gitea ?**
Exécutez la commande `gitea --version` dans un terminal et la version s'affichera.

**Question : Comment mettre à jour Gitea sans tout reconfigurer ?**
Tout d'abord, il est recommandé de faire une backup de la version acutelle pour éviter toute perte de donnée.

Les commandes suivantes sont les commandes recommandées par la documentation.

Avec l'utilisateur git créé précédemment, rendez-vous dans le dossier contenant le fichier gitea et executez :

```
./gitea dump -c /path/to/app.ini
```

Des sorties similaires devraient apparaître

```
2016/12/27 22:32:09 Creating tmp work dir: /tmp/gitea-dump-417443001
2016/12/27 22:32:09 Dumping local repositories.../home/git/gitea-repositories
2016/12/27 22:32:22 Dumping database...
2016/12/27 22:32:22 Packing dump files...
2016/12/27 22:32:34 Removing tmp work dir: /tmp/gitea-dump-417443001
2016/12/27 22:32:34 Finish dumping in file gitea-dump-1482906742.zip
```

La configuration étant désormais sécurisée, nous pouvons continuer pour la mise à jour. Il est très simple de mettre à jour :

- Téléchargez la nouvelle version sur https://dl.gitea.com/gitea/
- Modifiez le fichier `/usr/local/bin/gitea` par le fichier que vous venez de télécharger
- Relancez Gitea

**Gitea est désormais à jour.**

En cas de problème, vous pouvez revenir sur la backup avec les commandes suivantes :

```
unzip gitea-dump-1610949662.zip
cd gitea-dump-1610949662
mv app.ini /etc/gitea/conf/app.ini
mv data/* /var/lib/gitea/data/
mv log/* /var/lib/gitea/log/
mv repos/* /var/lib/gitea/data/gitea-repositories/
chown -R gitea:gitea /etc/gitea/conf/app.ini /var/lib/gitea
```

[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