

Rapport SAE2.03 : Installation de services réseaux

Renard Yann, Mekki Yanis

Février 2024

Sommaire

Introduction	1
I. Configuration matérielle dans VirtualBox	1
II. Installation de l'OS de base	2
III. Préparation du système	3
A. Accès sudo pour user	3
B. Installation des suppléments invités	3
IV. À propos de la distribution Debian	4
V. Installation préconfigurée	6
VI. Analyse préliminaire de git et des outils graphiques associés	7
A. Configuration globale de git	7
B. Les interfaces graphiques pour Git	7
C. Installons autre chose et comparons	8

Introduction

Ce document présente le travail réalisé durant la première semaine de la SAE 2.03. Il répond aux questions techniques posées dans le cadre de la configuration matérielle de la machine virtuelle sous VirtualBox. Ce rapport décrit **ce que nous avons fait et comment nous l'avons fait**, conformément aux consignes techniques fournies.

Récapitulatif général de la démarche : Nous avons lu attentivement les consignes fournies dans le document technique ainsi que les rapports des semaines précédentes. Nous avons reproduit les réponses attendues tout en respectant les paramètres techniques imposés. Ce rapport a été rédigé en Markdown avec une mise en forme optimisée pour une conversion facile via Pandoc en HTML et PDF.

I. Configuration matérielle dans VirtualBox

Dans cette première étape, nous avons configuré une machine virtuelle sous VirtualBox pour installer un environnement Debian fonctionnel. Nous avons commencé par lancer VirtualBox et créer une nouvelle machine virtuelle en sélectionnant l'option **Debian 64-bit**. Après avoir défini le nom de la VM comme **sae203**, nous avons ajusté les paramètres en attribuant **2048 Mo de RAM** et un **disque dur de 20 Go**, en choisissant une allocation dynamique pour optimiser l'espace de stockage.

Nous avons ensuite vérifié la configuration réseau par défaut en nous rendant dans les **paramètres de la VM**, sous l'onglet **Network**. Nous avons constaté que VirtualBox utilise par défaut le mode **NAT (Network Address Translation)**, ce qui permet à la machine virtuelle d'accéder à Internet via la connexion de l'hôte sans nécessiter de configuration spécifique. Pour s'assurer que tous les paramètres étaient corrects, nous avons localisé le fichier XML contenant la confi-

guration de la VM dans le répertoire `/usr/local/virtual_machine/infoetu/prenom.nom.etu`, nommé **sae203.vbox**. Ce fichier contient toutes les informations essentielles concernant la machine. Pour ajuster la puissance de la VM, nous avons modifié manuellement ce fichier en ouvrant un éditeur de texte et en remplaçant la ligne `<CPU>` par `<CPU count="2">`, ce qui nous a permis d'attribuer deux processeurs à la machine. Après avoir enregistré ces modifications, nous avons redémarré la VM.

Q&A 1. Que signifie “64-bit” dans “Debian 64-bit” ?

Une distribution 64-bit est conçue pour des processeurs capables de gérer des instructions sur 64 bits, ça permet une gestion de la mémoire et des calculs plus efficace qu'en 32 bits.

2. Quelle est la configuration réseau utilisée par défaut ?

Par défaut, VirtualBox configure le réseau en mode NAT (Network Address Translation), qui permet à la machine virtuelle d'accéder à Internet via la connexion de l'hôte.

3. Quel est le nom du fichier XML contenant la configuration de votre machine ?

Le fichier XML se nomme généralement « `login.vbox` » (porte le nom choisi pour la VM) et se trouve dans le répertoire de la machine virtuelle.

4. Sauriez-vous le modifier directement pour mettre 2 processeurs à votre machine ?

Oui. Il suffit d'éditer le fichier XML et de modifier la balise correspondante (par exemple, en passant de `<CPU count="1">` à `<CPU count="2">`), puis de sauvegarder le fichier pour que VirtualBox prenne en compte le changement.

II. Installation de l'OS de base

Une fois la machine virtuelle configurée, nous avons procédé à l'installation du système d'exploitation Debian. Nous avons commencé par télécharger l'image ISO de Debian en version **64-bit** depuis le [site officiel](#). Ensuite, nous avons monté cette image en tant que **lecteur CD virtuel** dans VirtualBox et démarré la VM. L'installateur Debian s'est lancé automatiquement, nous permettant de choisir les options de configuration adaptées à notre projet.

Durant l'installation, nous avons sélectionné **le français** comme langue principale et laissé les paramètres réseau en configuration automatique. Lors du choix des logiciels à installer, nous avons opté pour l'environnement de bureau **MATE** plutôt que **GNOME**, afin de privilégier un environnement graphique plus léger, mieux adapté aux machines virtuelles, et pour retranscrire notre environnement présent en salle TP. Nous avons également installé un **serveur web** et un **serveur SSH** pour permettre une gestion à distance de la machine. Après l'installation du système, nous avons redémarré la VM et vérifié que tout fonctionnait correctement (surtout si GRUB était bien configuré). Enfin, nous avons retiré l'ISO pour libérer de l'espace disque.

Q&A 1. Qu'est-ce qu'un fichier iso bootable ?

C'est une image disque regroupant l'ensemble des fichiers nécessaires pour démarrer et installer un système d'exploitation, reproduisant le contenu d'un support physique (DVD ou clé USB).

2. Qu'est-ce que MATE ? GNOME ?

MATE et GNOME sont des environnements de bureau pour Linux. GNOME offre une interface moderne et épurée, tandis que MATE est issu de la continuation de GNOME 2, apprécié pour sa légèreté et sa simplicité.

3. Qu'est-ce qu'un serveur web ?

Un serveur web est un logiciel (ou matériel) qui stocke, traite et distribue des pages web via le protocole HTTP, rendant les contenus accessibles aux navigateurs clients.

4. Qu'est-ce qu'un serveur ssh ?

C'est un service qui permet d'établir des connexions sécurisées à distance grâce au chiffrement, facilitant ainsi l'administration et l'accès sécurisé aux systèmes.

5. Qu'est-ce qu'un serveur mandataire ?

Également appelé proxy, il sert d'intermédiaire entre un client et un serveur. Il peut filtrer, sécuriser ou mettre en cache les échanges réseau afin d'améliorer la gestion du trafic et la sécurité.

III. Préparation du système

Une fois Debian installé, nous avons préparé le système en configurant les droits administratifs et en installant les suppléments invités.

A. Accès sudo pour user

Par défaut, l'utilisateur principal de Debian n'a pas accès aux privilèges **sudo**, ce qui nous a contraints à le configurer manuellement. Pour ce faire, nous avons ouvert un terminal et nous sommes connectés en tant que **root** en utilisant la commande **su -**. Ensuite, nous avons ajouté notre utilisateur au groupe **sudo** avec la commande :

```
usermod -aG sudo user
```

Afin que les modifications prennent effet, nous avons redémarré la session. Pour vérifier si l'utilisateur appartenait bien au groupe **sudo**, nous avons utilisé la commande **groups user**, qui nous a bien confirmé son appartenance à ce groupe.

Q&A Comment ajouter un utilisateur au groupe sudo ?

Pour ajouter un utilisateur au groupe sudo, on peut utiliser la commande **usermod -aG sudo user**. Pour voir si c'est effectif, soit on se déconnecte et reconnecte, soit on utilise directement la commande suivante **newgrp sudo**

Comment peut-on savoir à quels groupes appartient l'utilisateur user ?

On peut utiliser les commandes **groups user** ou **id user** dans le terminal. Ces commandes affichent la liste des groupes dont l'utilisateur fait partie.

B. Installation des suppléments invités

Pour améliorer l'expérience utilisateur et optimiser l'intégration de la VM avec l'hôte, nous avons installé les **suppléments invités**. Ils permettent d'améliorer la gestion de la souris, du clavier et de prendre en charge le redimensionnement dynamique de la fenêtre. Nous avons commencé par insérer l'ISO des **Guest Additions** en allant dans "Devices" > "Insert Guest Additions CD Image...". Ensuite, nous avons ouvert un terminal et monté manuellement le CD-ROM avec la commande :

```
sudo mount /dev/cdrom /mnt
```

Une fois l'image montée, nous avons lancé l'installateur des suppléments invités via la commande :

```
sudo /mnt/VBoxLinuxAdditions.run
```

Une fois l'installation terminée, nous avons redémarré la VM. Pour vérifier que les suppléments étaient bien activés, nous avons redimensionné la fenêtre de VirtualBox et constaté que l'affichage s'ajustait automatiquement, prouvant que l'installation avait réussi.

L'installation des *Guest Additions* a semé une confusion chez nous, le montage manuel du CD virtuel (`sudo mount /dev/cdrom /mnt`) et l'exécution du script d'installation (`VBoxLinuxAdditions.run`) exigeaient une manipulation précise des droits root via `sudo`. Ce que nous avons totalement oublié de faire et nous a coûté quelques heures de réflexions.

Q&A 1. Quel est la version du noyau Linux utilisé par votre VM ?

La version du noyau peut être obtenue en exécutant la commande `uname -r`. Dans notre VM on a `6.1.0-31-amd64` qui correspond à la version du noyau en cours d'utilisation.

2. À quoi servent les suppléments invités ? Donner 2 principales raisons de les installer.

Les suppléments invités permettent :

- Un redimensionnement dynamique de la fenêtre de la machine virtuelle pour une meilleure adaptation de l'affichage.
- Une intégration améliorée de la souris et du clavier entre l'hôte (nous) et la VM.

3. À quoi sert la commande `mount` (dans notre cas et en général) ?

Dans ce contexte, `mount` permet de monter le CD contenant les additions invitées pour accéder à son contenu. De manière générale, la commande attache un système de fichiers à un point de montage dans l'arborescence, ce qui rend son contenu accessible au système (souvent utilisé pour manipuler des ISO comme celui d'Arch qui nécessite cette étape pour le partitionnement).

IV. À propos de la distribution Debian

Enfin, nous avons approfondi nos connaissances sur Debian en nous intéressant à son fonctionnement et à son organisation. Nous avons découvert que le **Projet Debian** est une initiative communautaire visant à fournir un système d'exploitation libre et robuste. Le nom **Debian** provient de la combinaison des prénoms **Debra** et **Ian Murdock**, ses créateurs.



Concernant la maintenance des versions, nous avons appris qu’une version stable de Debian bénéficie d’un support **standard de 3 ans**, suivi d’un support **LTS (Long-Term Support) de 2 ans supplémentaires**. Certaines versions bénéficient d’un support encore plus long grâce au programme **ELTS (Extended Long-Term Support)**. Actuellement, Debian maintient trois versions actives : **stable**, **testing** et **unstable**. Chacune de ces versions joue un rôle spécifique dans le cycle de développement du système.

Nous avons également découvert l’origine des noms de code des différentes versions de Debian. Chaque version porte un nom tiré des personnages du film **Toy Story**, une tradition qui remonte aux débuts du projet. Par exemple, Debian 12 porte le nom **Bookworm**, tandis que les versions précédentes se nommaient **Bullseye**, **Buster**, ou même **Buzz** (étant le cosmonaute Buzz Lightyear bien sûr).

Pour finir nous avons étudié les architectures supportées par Debian. Actuellement, Debian est compatible avec une **large variété de processeurs**, dont les plus courants sont **amd64**, **i386**, **arm64**, **armhf**, mais aussi des architectures plus spécifiques comme **mips**, **mipsel**, **ppc64el** et **s390x**.

Q&A Pour ces questions, il est recommandé de consulter la documentation officielle (ex. : [debian.org](https://www.debian.org)).

1. Qu’est-ce que le Projet Debian ? D’où vient le nom Debian ?

Le Projet Debian est une organisation communautaire qui développe et maintient une distribution Linux libre et collaborative. Le nom “Debian” provient de la combinaison du prénom de Debra (une des fondatrices) et du nom de son partenaire, Ian Murdock.

2. Quelles sont les durées de prise en charge des versions (support standard, LTS et ELTS) ?

En général, une version Debian bénéficie d’un support standard pendant environ 3 ans, auquel s’ajoutent 2 années de support LTS (5 ans). Certaines versions peuvent bénéficier d’un support étendu (ELTS) qui prolonge cette période (souvent jusqu’à 6 ou 7 ans).

3. Pendant combien de temps les mises à jour de sécurité seront-elles fournies ?

Les mises à jour de sécurité sont assurées pendant toute la durée du support officiel d’une version, c’est-à-dire pendant le support standard et le support LTS (et éventuellement pendant l’ELTS, le cas échéant).

4. Combien de versions au minimum sont activement maintenues par Debian ? Donnez leur nom générique.

Debian maintient généralement trois versions actives :

- La version stable (pour les utilisateurs finaux),
- La version testing (en préparation pour devenir stable),
- La version unstable (destinée au développement continu).

5. D’où viennent les noms de code donnés aux distributions ?

Les [noms de code de Debian](#) sont choisis par la communauté proviennent des personnages des films « Toy Story » par Pixar. Par exemple, « [bookworm]([Debian – Debian “bookworm” Release Information](#)) » pour Debian 12 était un ver de terre vert équipé d’un flash et qui adore lire des livres.

6. Combien et lesquelles sont les architectures prises en charge par Debian Bullseye ?

Debian Bullseye prend officiellement en charge environ 8 architectures, parmi lesquelles on retrouve :

Architecture	Description
amd64	Architecture 64 bits pour processeurs AMD et Intel
i386	Architecture 32 bits pour processeurs Intel/AMD
arm64	Architecture 64 bits pour processeurs ARM
armhf	Architecture ARM 32 bits avec unité de calcul flottant
mips	Architecture MIPS (big-endian)
mipsel	Architecture MIPS (little-endian)
ppc64el	Architecture PowerPC 64 bits (little-endian)
s390x	Architecture IBM System zag-0-1ikica249ag-1-1ikica249

7. Première version avec un nom de code

- *Quel a été le premier nom de code utilisé ?*
Le premier nom de code utilisé est « buzz ».
- *Quand a-t-il été annoncé ?*
Il a été annoncé au début des années 1990 (la première version stable « Debian 1.1 » a été publiée en 1996).
- *Quelle était le numéro de version de cette distribution ?*
Il s'agissait de Debian 1.1.

8. Dernier nom de code attribué

- *Quel est le dernier nom de code annoncé à ce jour ?*
Le dernier nom de code est « bookworm » pour Debian 12 ou « trixie » pour Debian 13 et « sid » désigne la version instable.
- *Quand a-t-il été annoncé ?*
Il a été annoncé en 2022.
- *Quelle est la version de cette distribution ?*
Il correspond à Debian 12.

V. Installation préconfigurée

Pour cette étape, nous avons automatisé l'installation de Debian avec des fichiers de pré-configuration. Après avoir créé une nouvelle machine virtuelle, nous utilisons l'archive d'auto-installation décompressée dans `/usr/local/virtual_machine/infoetu/prenom.nom.etu`. Le fichier `S203-Debian12.viso` a été modifié pour remplacer `@@UUID@@` avec cette commande :

```
sed -i -E "s/(--iprt-iso-maker-file-marker-bourne-sh).*/\1=$(cat /proc/sys/kernel/random/uuid)/" S203-Debian12.viso
```

Nous l'avons inséré comme lecteur optique et lancé l'installation, qui s'est déroulée automatiquement rt comme prévu. Pour répondre aux exigences, nous avons ajusté `preseed.cfg` :

- **Droits sudo** : `d-i preseed/late_command string in-target usermod -aG sudo user`
- **MATE** : `d-i pkgssel/include string mate-desktop-environment`
- **Paquets** : `d-i pkgssel/include string sudo git sqlite3 curl bash-completion neofetch`

Après relance, nous avons vérifié avec `sudo -l`, `mate-about`, et `neofetch`. Les additions invités fonctionnent (redimensionnement OK). La VM est prête pour la suite.

```
55  ## Installation meta-paquetages
56  tasksel tasksel/first multiselect standard ssh-server mate-desktop
57
58  ### Installation des paquets supplémentaires
59  d-i pkgsel/include string sudo git sqlite3 curl bash-completion neofetch
60
61  ### Mise à jour des paquets (optionnel)
62  d-i pkgsel/upgrade select full-upgrade
63  d-i pkgsel/update-policy select none
64
65  ### Suivi statistiques paquets installés
66  popularity-contest popularity-contest/participate boolean false
67
68  ### Grub
69  d-i grub-installer/grub2_instead_of_grub_legacy boolean true
70  d-i grub-installer/only_debian boolean true
71  d-i grub-installer/timeout string 2
72  d-i grub-installer/bootdev string default
73
74  d-i finish-install/reboot_in_progress note
75
```

FIGURE 1 – pressed.cfg

VI. Analyse préliminaire de git et des outils graphiques associés

Nous partons d’une machine virtuelle Debian 12 (sae203) avec MATE et Git déjà installés via une pré-configuration (`preseed.cfg`). Cette section détaille la configuration de Git et l’évaluation de ses outils graphiques.

A. Configuration globale de git

Nous avons paramétré Git pour notre utilisateur en ouvrant un terminal et en exécutant :

```
git config --global user.name "Prénom Nom"
# git config --global user.name "Yann Renard"
git config --global user.email "prenom.nom@exemple.com"
# git config --global user.email "yannrenard1025@gmail.com"
git config --global init.defaultBranch "master"
# ou > git config --global init.defaultBranch "main"
```

- La première commande associe notre identité aux commits.
- La deuxième lie une adresse email.
- La troisième définit “master” comme branche par défaut, évitant un avertissement lors de l’initialisation d’un dépôt.

B. Les interfaces graphiques pour Git

Nous avons installé les paquets `gitk` et `git-gui` avec la commande :


```
sudo apt install gitk git-gui
```

Q&A Qu'est-ce que le logiciel gitk ? Comment se lance-t-il ?

Gitk est un visualiseur d'historique Git qui affiche les commits sous forme d'arbre graphique (comme beaucoup d'interfaces le font avec Git en réalité). Il se lance via la commande `gitk` dans un dépôt Git existant. Nous l'avons testé avec un dossier contenant notre [SAE de développement iJava](#) hébergé sur Github, observant les commits et les branches.

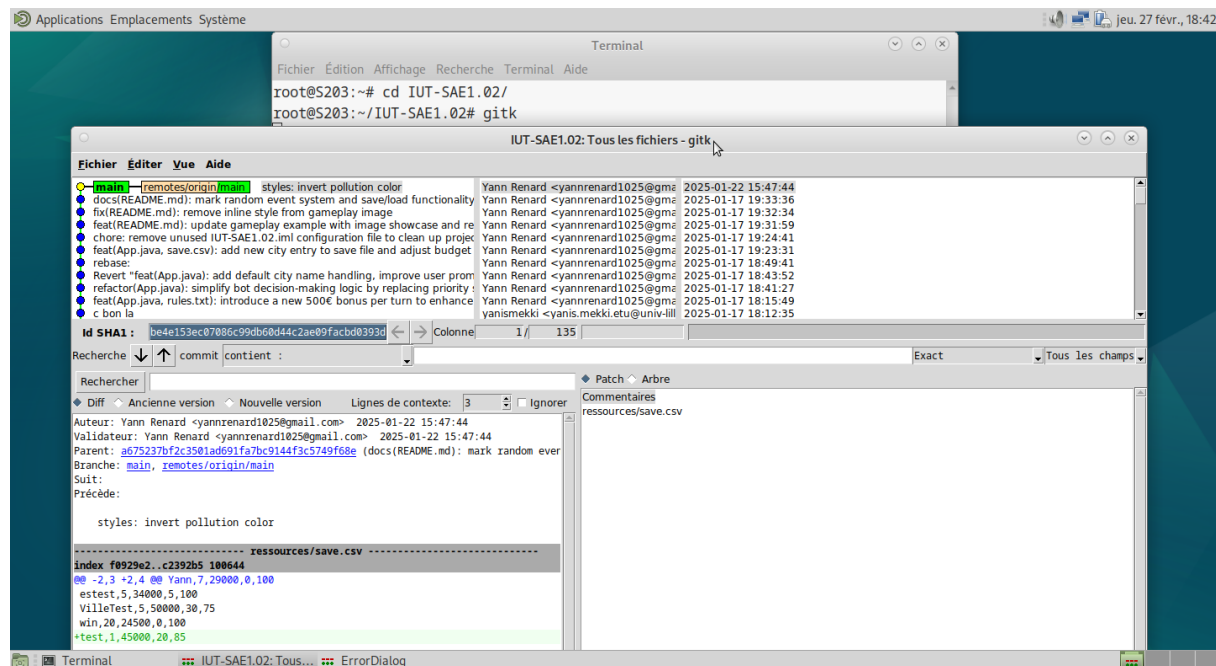


FIGURE 2 – GitK

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

Git-gui est une interface pour créer, modifier et gérer des commits de manière graphique. Il se lance avec `git gui` dans un répertoire Git. Nous l'avons utilisé pour ajouter et valider des fichiers texte.

C. Installons autre chose et comparons

Nous avons choisi **GitKraken**, **LazyGit**, et **gitg** comme alternatives, toutes gratuites pour un usage personnel (on a légèrement abusé sur le nombre de soft mais ils sont tout trois différents et vous allez comprendre pourquoi).

— Installation :

- **GitKraken** : Téléchargé depuis gitkraken.com (.deb) et installé avec :

```
sudo dpkg -i gitkraken-amd64.deb
sudo apt install -f
```

- **LazyGit** : Installé via Homebrew (indisponible directement sur Debian malheureusement) avec :

```
brew install lazygit
```

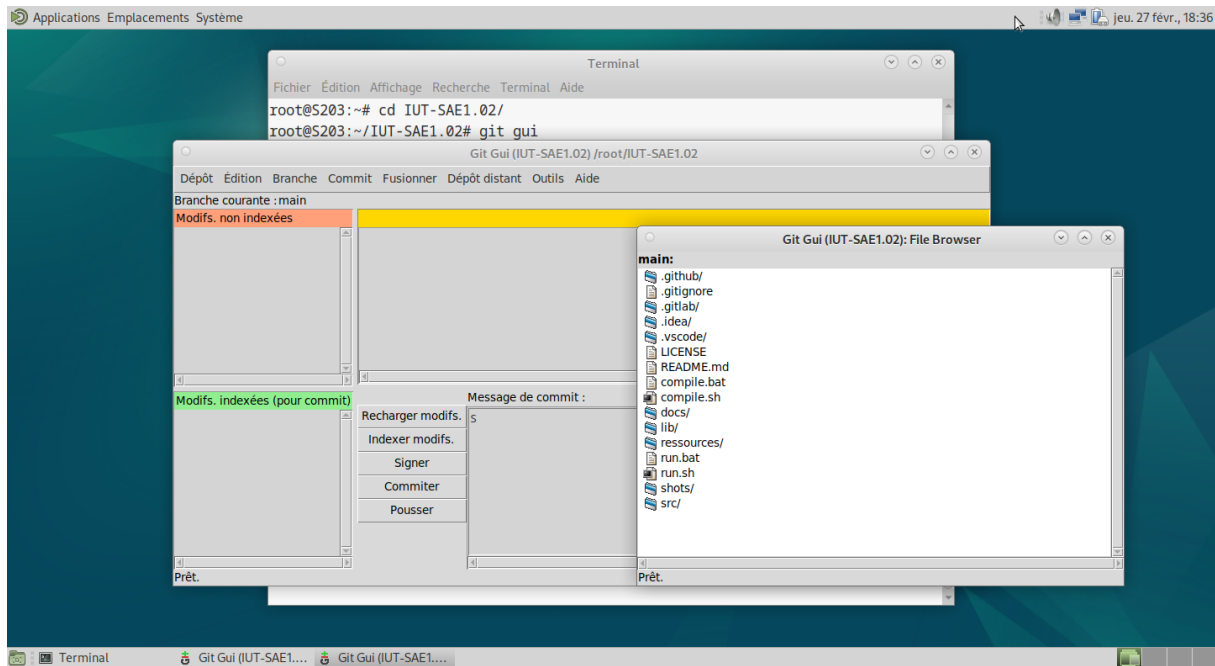



FIGURE 3 – GitGUI

- **gitg** : Installé via Flathub et Flatpak avec :

```
flatpak install flathub org.gnome.gitg
```
- **Pourquoi ce choix ?**
 - **GitKraken** : Interface moderne et intuitive.
 - **LazyGit** : TUI rapide et clavier-centré.
 - **gitg** : Léger et intégré à GNOME/MATE.
- **Comparaison :**
 - **Gitk** : Visualisation simple, rapide, mais très peu interactive.
 - **Git-gui** : Fonctionnel pour les commits, mais basique.
 - **GitKraken** : Riche (branches, remotes), visuel, mais lourd. Avantage : ergonomie. Inconvénient : gourmand en ressources.
 - **LazyGit** : Efficace en terminal, tout au clavier. Avantage : vitesse. Inconvénient : courbe d'apprentissage élevée (suit le principe de NeoVim).
 - **gitg** : Minimaliste, idéal pour consulter l'historique. Avantage : légèreté. Inconvénient : moins de fonctions.
 - **Ligne de commande** : Contrôle total, mais sans aide visuelle.

Nous avons testé avec un dépôt de fichiers texte et le dépôt de notre [SAE1.02 de développement](#), préférant GitKraken pour son interface et LazyGit pour sa rapidité.