
Rapport final SAÉ 2.03: Installation de services réseaux

3-23-2024

Table des matières

1	Préparation d'une machine virtuelle	2
1.1	Préparation de la machine virtuelle	2
1.1.1	Caractéristiques de la machine virtuelle:	2
1.1.2	Questions:	2
1.2	Installation de l'OS	3
1.2.1	Caractéristiques de l'installation Debian:	4
1.2.2	Questions:	4
2	Préparation du système	5
2.1	Accès sudo pour <i>user</i>	5
2.1.1	Questions:	6
2.2	Installation des suppléments invités	6
2.2.1	Questions:	7
3	À propos de la distribution Debian	7
3.0.1	Questions:	7
4	Installation préconfigurée	9
5	Analyse préliminaire de git et des outils graphiques associés	11
5.1	Configuration globale de git	11
5.2	Les interfaces graphiques pour git	11
5.2.1	Questions:	12
5.3	Installons autre chose et comparons	13
5.3.1	Questions:	13
6	Installation de Gitea	14
6.1	Préliminaire: redirection de port	14
6.2	Installation de Gitea	15
6.2.1	Questions:	15
6.3	Installation du binaire	15
6.4	Mise à jour du binaire de service Gitea	21
6.4.1	Questions:	21
6.5	Tests d'utilisation	22
7	Sources et références	22

1 Préparation d'une machine virtuelle

1.1 Préparation de la machine virtuelle

Pour cette étape, nous avons du configurer une machine virtuelle via le logiciel de virtualisation **VirtualBox**.

1.1.1 Caractéristiques de la machine virtuelle:

Nom	sae203
Type	Linux
Version	Debian 12 (64-bit)
Mémoire vive	2048 Mo
Espace disque	20 Go

1.1.2 Questions:

Q: Que signifie "64-bit" dans "Debian 64-bit" ?

A: Le terme "64-bit" représente une architecture système pouvant traiter des données par blocs de 64 bits.

Q: Quelle est la configuration réseau utilisée par défaut ?

A:

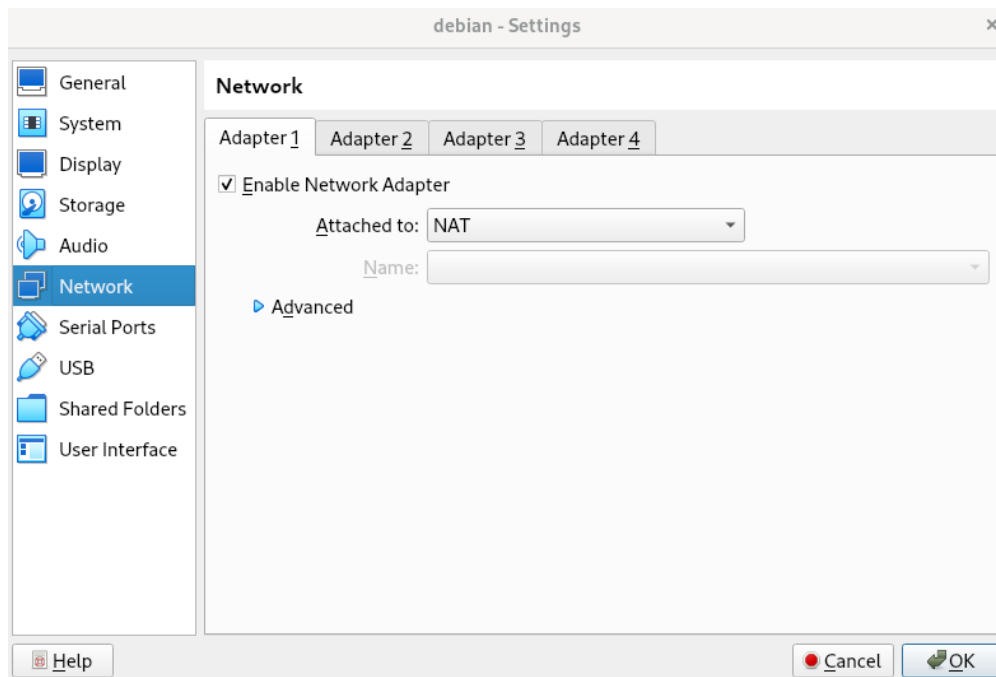


Figure 1: Configuration réseau standard sur VirtualBox

Q: Quel est le nom du fichier XML contenant la configuration de votre machine ?

A: Son nom est `sae203.vbox`.

Q: Sauriez-vous modifier directement ce fichier de configuration pour mettre 2 processeurs à votre machine ?

A: Pour cela, il suffit de modifier la valeur dans la ligne `<CPU count="x">`, avec x étant le nombre de coeurs à allouer. Dans ce cas, cette ligne de configuration deviendrait:

```
1 <CPU count="2">
```

1.2 Installation de l'OS

Après avoir configuré la machine virtuelle, nous devons installer sur celle-ci **Debian** (ici la version 12, nom de code "Bookworm"), avec l'environnement graphique **MATE**, et quelques logiciels supplémentaires.

Nous avons pour ceci utilisé l'ISO d'installation de Debian `netinst`, pour **Network Installer**, qui requiert un accès réseau.

1.2.1 Caractéristiques de l'installation Debian:

- **Nom de la machine:** *serveur*
- **Domaine:** *[vide]*
- **Pays/langue:** *France*
- **Miroir:** *http://debian.polytech-lille.fr*
- **Proxy:** *[vide]*
- **Compte administrateur:** *root / root*
- **Compte utilisateur:** *user / user*
- **Partitionnement:** *Une seule partition de la taille du disque*
- **Sélection des logiciels de démarrage:**
 - *Environnement de bureau Debian*
 - * **MATE**
 - *Serveur web*
 - *Serveur SSH*
 - *Utilitaires usuels du système*

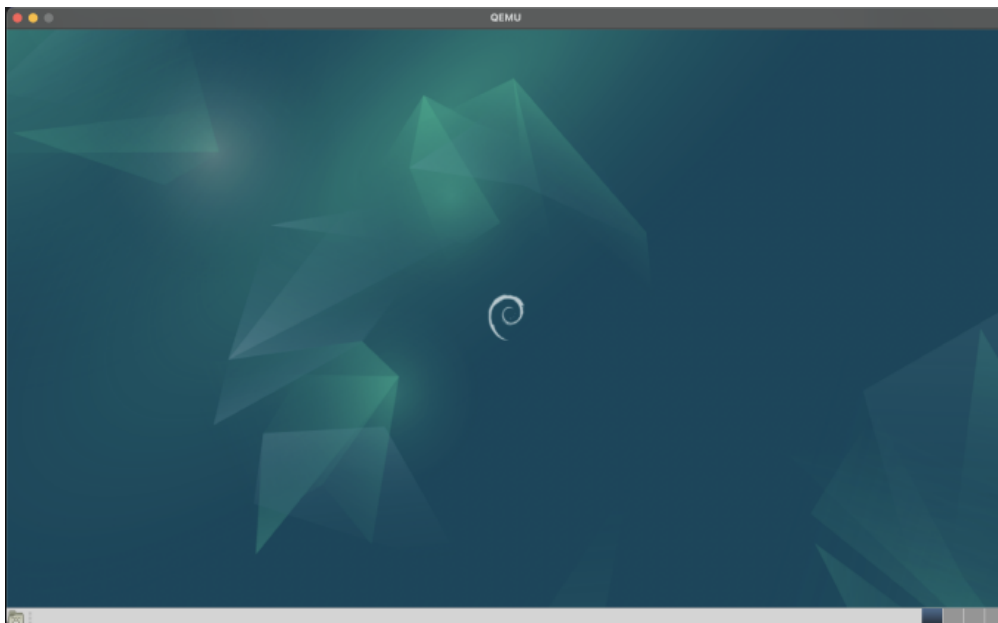


Figure 2: Configuration simple de Debian 12 avec MATE

1.2.2 Questions:

Q: *Qu'est-ce qu'un fichier iso bootable ?*

A: Un fichier ISO (en référence à la norme internationale **ISO 9660** qui définit les systèmes de fichiers sur CD et DVD) est un type de fichier représentant une copie du contenu d'un CD ou d'un DVD, contenant des fichiers binaires permettant par exemple d'installer un système d'exploitation sur un ordinateur.

Q: Qu'est-ce que **MATE** ? **GNOME** ?

A: **MATE** et **GNOME** sont deux environnements de bureaux pour les systèmes Linux.

Q: Qu'est-ce qu'un serveur web ?

A: Un serveur web est une machine qui stocke des fichiers d'un site web (HTML, CSS...), et interagit avec les autres machines sur un réseau (public ou privé) via le protocole **HTTP(S)**.

Q: Qu'est-ce qu'un serveur ssh ?

A: Un serveur **SSH** (pour **Secure Shell Protocol**), permet à des utilisateurs de se connecter à distance à une machine (avec comme interface un shell de commandes) via un protocole de connection sécurisé.

Q: Qu'est-ce qu'un serveur mandataire ?

A: Un serveur mandataire, ou **proxy**, est une machine qui sert d'intermédiaire dans les communications de deux autres machines.

2 Préparation du système

2.1 Accès sudo pour user

La commande **sudo** permet à un utilisateur (avec les bonnes permissions) d'exécuter des commandes en tant qu'administrateur.

Pour ajouter les permissions administrateur à notre utilisateur *user*, nous allons nous connecter en **tty** (affichage textuel) avec la suite de touches **Ctrl + Alt + 1** à l'utilisateur *root*, puis exécuter la commande suivante:

```
1 # Ajoute l'utilisateur user dans le groupe sudo
2 usermod -aG sudo user
```

Q: Comparez-le aux outils inclus avec *git* (et installé précédemment) ainsi qu'avec ce qui serait fait en ligne de commande pure : fonctionnalités, avantages, inconvénients... - **usermod**: Commande qui permet de modifier les groupes d'un utilisateur. - **-a**: Option pour ajouter un ou plusieurs groupes. - **-G**: Option pour spécifier les groupes à ajouter. - **sudo**: Le groupe (par défaut sur Debian) permettant à un utilisateur d'exécuter des commandes en tant qu'administrateur. - **user**: Le nom de notre utilisateur.

Il suffit de par la suite se déconnecter, revenir à l'affichage graphique avec la suite de touches **Ctrl** + **Alt** + **6**, puis de se connecter avec notre utilisateur *user*.

En lançant un terminal, nous pouvons vérifier si notre utilisateur possède bien les permissions administrateur:

```
1 user@server:~$ sudo echo test
2 [sudo] password for user:
3 test
4 user@server:~$
```

2.1.1 Questions:

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

A: Pour connaître les groupes d'un utilisateur, on utilise la commande **groups**, ici **groups user**.

2.2 Installation des suppléments invités

Pour améliorer le confort d'utilisation de Linux sur VirtualBox, il faut installer des programmes supplémentaires depuis une image ISO.

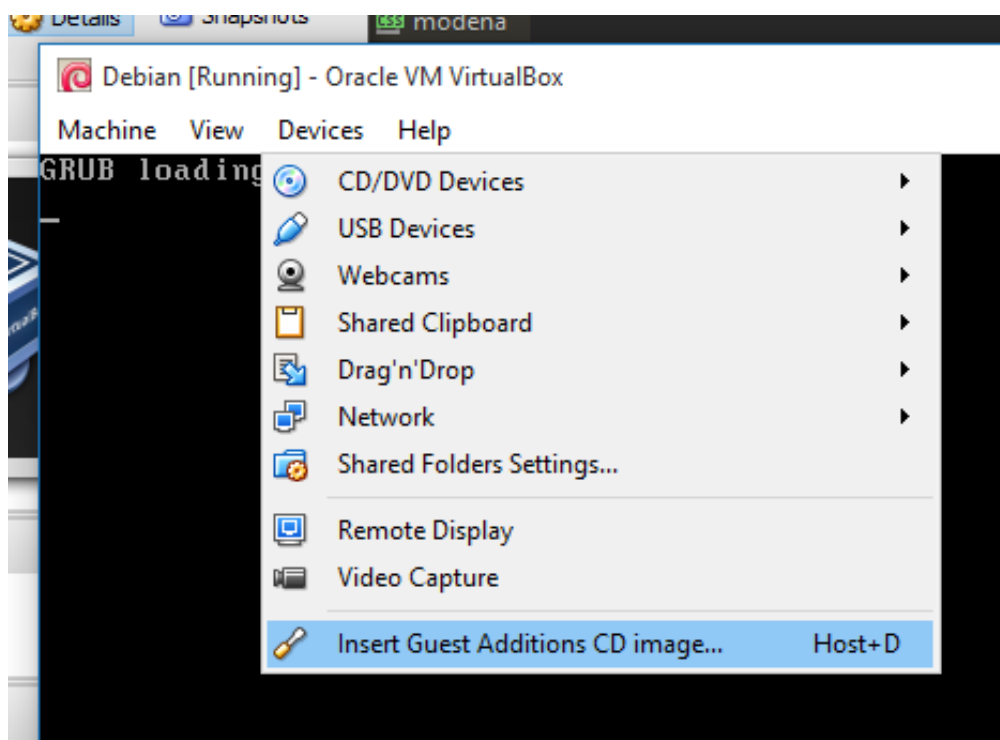


Figure 3: Option pour les CD de suppléments invités

Une fois le CD des suppléments invités inséré:

1. Monter le CD:

```
1 # Monte le CD de suppléments invités dans /mnt
2 sudo mount /dev/cdrom /mnt
```

2. Lancer le programme et installer les suppléments:

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

Une fois les suppléments installés, nous pouvons redémarrer la machine virtuelle.

2.2.1 Questions:

Q: *Quel est la version du noyau Linux utilisé par votre VM ? N'oubliez pas, comme pour toutes les questions, de justifier votre réponse.*

A: À l'aide de la commande `uname -rs` (-r pour la version, -s pour le nom), nous pouvons connaître la version du kernel Linux présent sur la machine virtuelle. Dans notre cas, il s'agit du kernel `Linux` classique, en version `6.1.0-18-amd64`.

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

A: Les suppléments invités est un ensemble de drivers de périphériques ainsi que d'applications système améliorant la performance et son utilisabilité. Ils intègrent par exemple des **dossiers partagés** entre l'hôte et la machine virtuelle, ainsi qu'un **meilleur support vidéo** à l'aide de drivers d'accélération 3D.

Q: *À quoi sert la commande `mount` (dans notre cas de figure et dans le cas général) ?*

A: La commande `mount` sert à monter et rendre accessible un système de fichiers (disque dur, clé USB, CD...) sur un répertoire donné en paramètre.

3 À propos de la distribution Debian

3.0.1 Questions:

Q: *Qu'est-ce que le Projet Debian ? D'où vient le nom Debian ?*

A: Le projet Debian est une association de développeurs indépendants qui ont pour but de fournir un système d'exploitation libre. Le nom **Debian** est une fusion du prénom du créateur de Debian: **Ian** Murdock, et celui de sa femme à l'époque: **Debrah**.

Q: Il existe 3 durées de prise en charge (support) de ces versions : la durée minimale, la durée en support long terme (LTS) et la durée en support long terme étendue (ELTS). Quelle sont les durées de ces prises en charge ?

A: La durée minimale de support est de:

- **Durée minimale:** 3 ans
- **LTS:** 5 ans
- **ELTS:** 10 ans

Q: Pendant combien de temps les mises à jour de sécurité seront-elles fournies ?

A: Les mises à jour de sécurité sont fournies jusqu'à **1 an** après la sortie de la version suivante.

Q: Combien de version au minimum sont activement maintenues par Debian ? Donnez leur nom générique (= les types de distribution).

A:

- **Debian 10** - "Buster" (2019)
- **Debian 11** - "Bullseye" (2021)
- **Debian 12** - "Bookworm" (2023, actuel)
- **Debian Unstable** - "Sid" (rolling-release)

Q: Chaque distribution majeure possède un nom de code différent. Par exemple, la version majeure actuelle (Debian 12) se nomme Bookworm. D'où viennent les noms de code données aux distributions ?

A: Les noms de code des versions de Debian viennent des noms des personnages de **Toy Story**.

Q: L'un des atouts de Debian fut le nombre d'architecture (= processeurs) officiellement prises en charge. Combien et lesquelles sont prises en charge par la version Bullseye ?

A: Debian 11 "Bullseye" supporte 9 architectures différentes:

- **AMD64**
- **AArch64**
- **ARMel**
- **ARMhf**
- **i386**
- **MIPS**
- **MIPS64**
- **POWER64**
- **IBM S/390**

Q: Quel a été le premier nom de code utilisé ?

A: Le premier nom de code utilisé par le Projet Debian est “**Buzz**”.

Q: *Quand a-t-il été annoncé ?*

A: Le nom de code ainsi que la version ont été annoncés (et sortis) le **17 juin 1996** sur la [mailing list de Debian](#).

Q: *Quel était le numéro de version de cette distribution ?*

A: Il s’agit de la version **1.1**.

Q: *Quel est le dernier nom de code annoncé à ce jour ?*

A: Le dernier nom de code annoncé par le Projet Debian est “**Forky**”.

Q: *Quand a-t-il été annoncé ?*

A: Le nom de code a été annoncé le **13 octobre 2022** sur la [mailing list de Debian](#).

Q: *Quelle est la version de cette distribution ?*

A: Le nom de code “**Forky**” correspond à la version **14** de Debian.

4 Installation préconfigurée

Pour cette partie, nous avons dû modifier un fichier de configuration pour automatiser une installation de Debian.

Après avoir téléchargé l’archive [autoinstall_Debian.zip](#) sur Moodle, il a d’abord fallu générer un UUID aléatoire sur le fichier de script [S203-Debian12.viso](#) avec la commande suivante:

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

- **sed**: Pour “**stream editor**”, c’est un programme UNIX permettant d’appliquer des modifications à des fichiers.
- **-i**: Option pour modifier des fichiers.
- **-E**: Option permettant l’utilisation d’expressions régulières.
- **s/(--iprt-iso-maker-file-marker-bourne-sh).*\$/\1=**: Sélection de la partie à modifier.
- **cat**: Commande UNIX pour afficher le contenu d’un fichier.
- **/proc/sys/kernel/random/uuid**: Fichier spécial sur Linux permettant de générer aléatoirement des UUID (**U**niversally **U**nique **I**dentifier).
- **S203-Debian12.viso**: Le nom du fichier à modifier.

Cette commande permet de remplacer tout ce qui se trouve après la ligne `--iprt-iso-maker-file-marker-bourne-sh=` du fichier **S203-Debian12.viso** par un UUID aléatoire.

Après avoir lancé une installation automatique sans rien modifier, on se retrouve avec un système Debian, cependant très peu ergonomique: pas d'affichage graphique, pas de programmes de développement présent, etc.

Pour remédier à cela, nous avons du modifier le fichier de configuration de l'installateur automatique Debian, ici `preseed-fr.cfg`.

Parmi les modifications demandées, on trouve:

- Ajout du groupe `sudo` à l'utilisateur standard
- Installation de l'environnement bureau **MATE**
- Ajout des paquets suivants:
 - `sudo`: Permet l'exécution de commandes en tant qu'administrateur.
 - `git`: Logiciel de gestion de versions.
 - `sqlite3`: Bibliothèque de gestion de bases de données.
 - `curl`: Utilitaire de transfert de données réseau.
 - `bash-completion`: Collections de scripts de complétion pour améliorer l'expérience utilisateur sur le shell `bash`.
 - `neofetch`: Logiciel permettant d'afficher nombreuses informations sur le système et la session en cours.

Pour ajouter les droits `sudo` à l'utilisateur créé pendant l'installation automatique, il suffit de rajouter **sudo** sur la ligne suivante:

```
d-i passwd/user-default-groups string audio cdrom video sudo
```

Pour ajouter l'environnement de bureau MATE, il suffit de rajouter cette ligne dans le fichier de configuration:

```
tasksel tasksel/first multiselect standard ssh-server mate-desktop
```

Enfin, pour ajouter les paquets supplémentaires, il suffit de rajouter cette ligne dans le fichier de configuration:

```
d-i pkgssel/include string sudo, git, sqlite3, curl, bash-completion, neofetch
```

Maintenant que le fichier de configuration est fini, nous pouvons le sauvegarder et recommencer l'installation automatisée pour voir si nos changements ont bien marché.

Une fois l'installation finie, nous pouvons redémarrer la machine virtuelle, et voir que tout a été installé correctement.

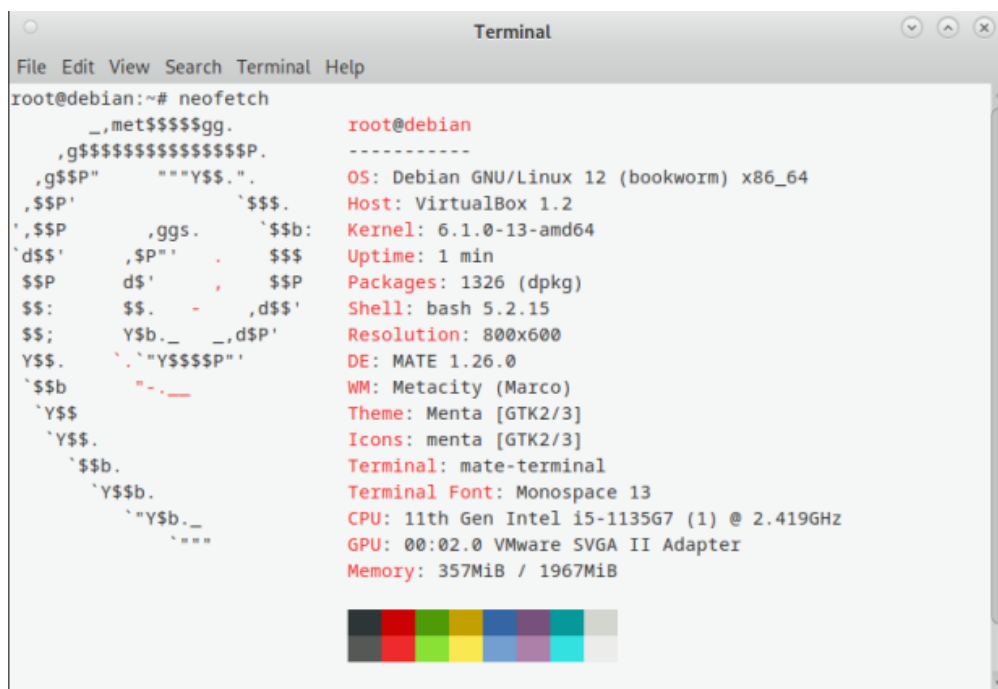


Figure 4: Affichage de neofetch sur Debian

5 Analyse préliminaire de git et des outils graphiques associés

5.1 Configuration globale de git

Nous avons d'abord du configurer notre profil git avec les commandes suivantes (exemple):

```
1 # Change le nom présent dans les commits git
2 git config --global user.name "[prénom] [NOM]"
3 # Change l'email présent dans les commits git
4 git config --global user.email "[adresse mail]"
5 # Change la branche par défaut pour la création de dépôts locaux git
6 git config --global init.defaultBranch "master"
```

5.2 Les interfaces graphiques pour git

Pour installer les deux interfaces graphiques de git sur Debian, il suffit d'utiliser les deux commandes suivantes:

```
1 # Installation de gitk et git-gui via APT
2 sudo apt install gitk
3 sudo apt install git-gui
```

5.2.1 Questions:

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

A: Le logiciel `gitk` est un logiciel graphique permettant la visualisation de l'historique des actions faites sur le dépôt. Pour le lancer, il suffit dans un terminal de se déplacer dans un dossier contenant un projet git à l'aide de la commande `cd`, puis d'exécuter la commande `gitk`.

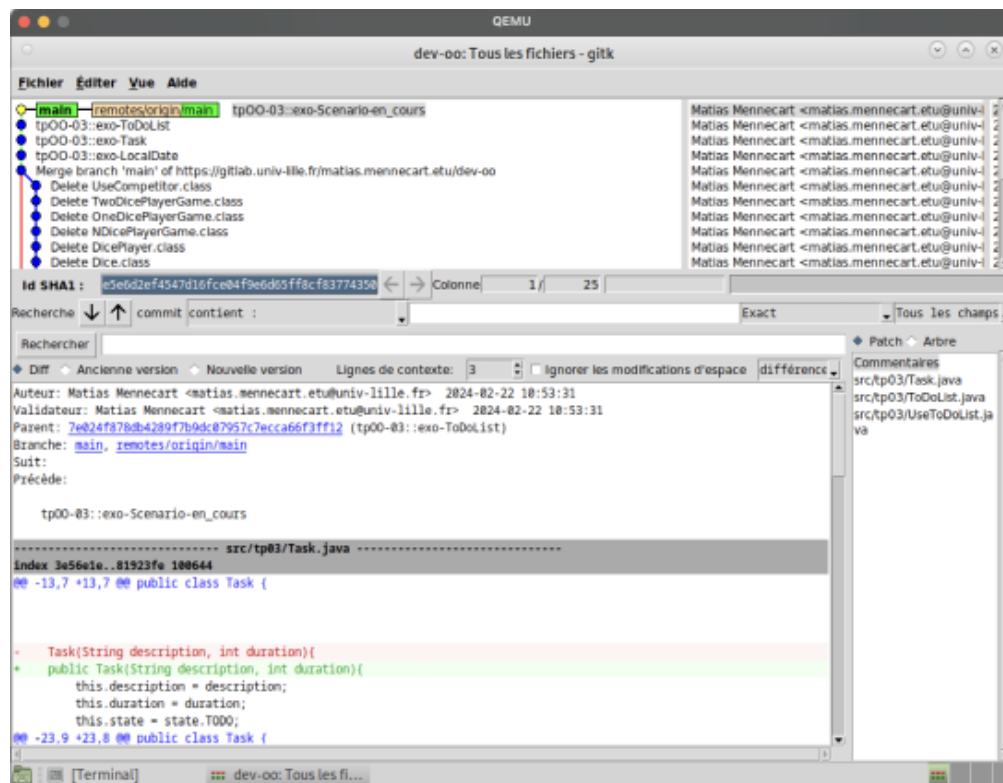


Figure 5: Capture d'écran de gitk

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

A: Le logiciel `git-gui` est une extension de git offrant une interface graphique pour la création de commits, de branches, d'actions fetch/push, etc. Pour le lancer, il suffit de simplement exécuter la commande `git gui` depuis un terminal ou autre lanceur de commandes, et l'interface graphique permettra à l'utilisateur de créer un nouveau dépôt ou d'en ouvrir un déjà existant.

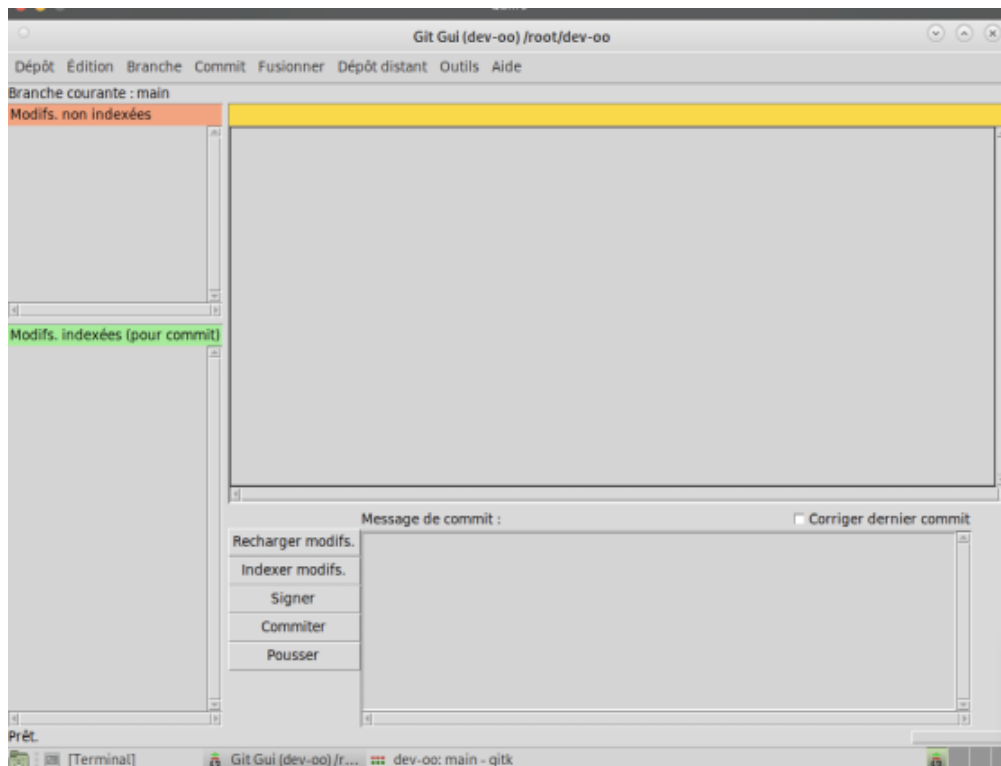


Figure 6: Capture d'écran de git-gui

5.3 Installons autre chose et comparons

5.3.1 Questions:

Q: Pourquoi avez-vous choisi ce logiciel ?

A: Nous avons choisi **Aurees**, car celui-ci est disponible sur tous les systèmes d'exploitation majeurs (Linux, macOS et Windows), ce qui le rend utilisable par tout le monde, sans contrainte, qui ne pose pas de problème en cas de changement de système, ainsi que pour son efficacité dans la résolution des conflits de merge.

Q: Comment l'avez-vous installé ?

A: Aurees possède des installateurs pour tout système d'exploitation majeur sur son [site web](#). N'étant pas présent par défaut sur les dépôts apt de Debian, nous avons dû télécharger le fichier **.deb** depuis le site, puis l'installer à l'aide de la commande suivante:

```
1 # Installation via DPKG (Debian) de Aurees
2 sudo dpkg -i AureesSetup-linux.deb
```

Q: Comparez-le aux outils inclus avec git (et installé précédemment) ainsi qu'avec ce qui serait fait en ligne de commande pure : fonctionnalités avantages, inconvénients...

A:

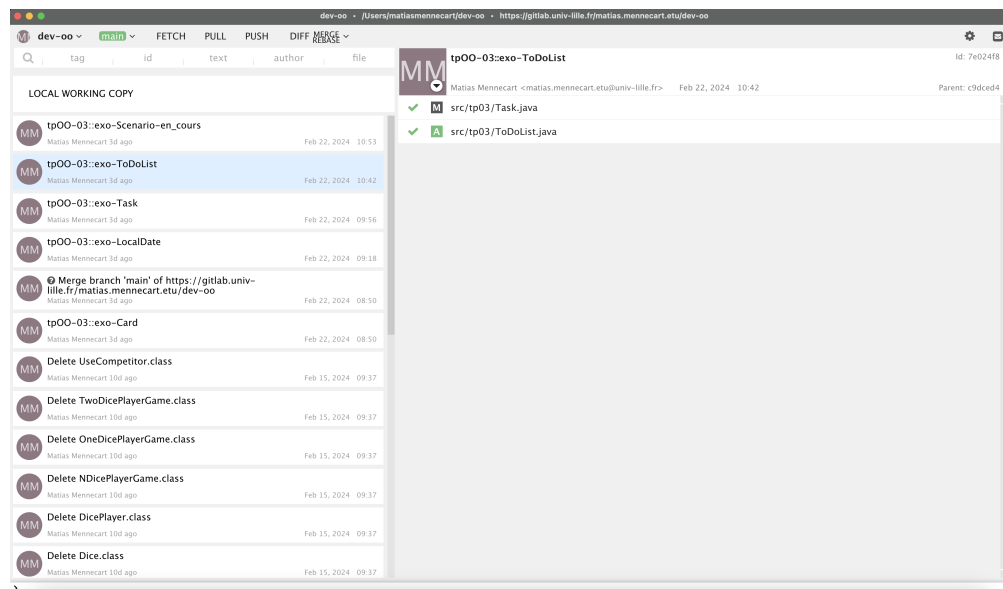


Figure 7: Capture d'écran de Aurees

6 Installation de Gitea

6.1 Préliminaire: redirection de port

Pour permettre à notre machine hôte de se connecter au service web de Gitea, qui sera installé sur la machine virtuelle, il faudra configurer la machine virtuelle pour rediriger le port utilisé par Gitea (ici, le port **3000**) vers la machine hôte.

Pour cela, il suffit de se rendre dans la configuration de la machine virtuelle sur VirtualBox, puis **Réseau** -> **Advanced** -> **Redirection de ports**.

Il suffit enfin d'ajouter la règle TCP de redirection du port 3000 dans la table et de sauvegarder.

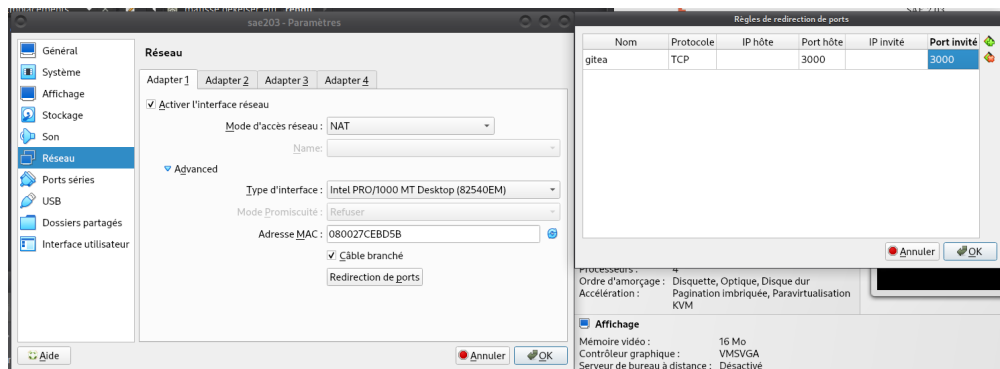


Figure 8: Paramètres de redirection de ports sur VirtualBox

6.2 Installation de Gitea

6.2.1 Questions:

Q: Qu'est-ce que Gitea ?

A: Gitea est un logiciel open-source, proposant un service d'hébergement de développement logiciel, utilisant le logiciel de gestion de versions **git**.

Q: À quels logiciels bien connus dans ce domaine peut-on le comparer (en citer au moins 2) ?

A: En services web git gratuits/open-source, permettant du self-host, on peut citer [GitLab](#) ou encore [Gogs](#). En service web git propriétaire, sans possibilité de self-hosting, on peut citer le très connu [GitHub](#).

6.3 Installation du binaire

Pour commencer l'installation de Gitea, nous allons d'abord télécharger la dernière version binaire disponible.

```
1 # Télécharge le fichier binaire
2 wget -O gitea https://dl.gitea.com/gitea/1.21.9/gitea-1.21.9-linux-amd64
3 # Rend le fichier exécutable
4 chmod +x gitea
```

Pour vérifier qu'il n'y a pas eu de problèmes avec le téléchargement du fichier, on compare son hash **SHA256** avec celui présent sur le site.

```
1 # Hash SHA256 du fichier local
2 sha256sum gitea
```



```
3 ffd269c8e6910fa21a38849d5155c953a8a83f364470d675e9378dbb3b231fc8 gitea
4
5 # Hash SHA256 fourni par Gitea
6 curl https://dl.gitea.com/gitea/1.21.9/gitea-1.21.9-linux-amd64.sha256
7 ffd269c8e6910fa21a38849d5155c953a8a83f364470d675e9378dbb3b231fc8 gitea
  -1.21.9-linux-amd64
```

Étant donné que les hash sont similaires, il n'y a pas eu de problème pendant le téléchargement, alors nous pouvons procéder à l'installation et à la configuration de Gitea.

Pour bien fonctionner, Gitea fonctionne à l'aide d'un utilisateur spécial `git` dédié. Sur Debian, on crée cet utilisateur avec la commande suivante:

```
1 sudo adduser \ # Commande de création d'utilisateur
2   --system \ # Utilisateur système
3   --shell /bin/bash \ # Shell bash par défaut
4   --gecos 'Git Version Control' \ # Informations GECOS pour /etc/
   passwd
5   --group \ # Création d'un groupe git
6   --disabled-password \ # Aucun mot de passe
7   --home /home/git \ # Création dossier home
8   git # Nom de l'utilisateur
```

Ensuite, il faut créer les dossiers nécessaires aux fichiers de Gitea (logs, db, config, etc.):

```
1 # Création des working directory de Gitea
2 sudo mkdir -p /var/lib/gitea/{custom,data,log}
3 # Changement des permissions owner dans /var/lib/gitea (récursif)
4 sudo chown -R git:git /var/lib/gitea/
5 # Changement récursif des droits de lecture
6 sudo chmod -R 750 /var/lib/gitea/
7 # Création du dossier pour les fichiers de configuration de Gitea
8 sudo mkdir /etc/gitea
9 # Changement des permissions owner (on ajoute le groupe git
   temporairement le temps de l'installation)
10 sudo chown root:git /etc/gitea
11 # Changement des droits de lecture
12 sudo chmod 770 /etc/gitea
```

Une fois les dossiers créés, nous pouvons déplacer l'exécutable de Gitea dans un dossier du **PATH**:

```
1 sudo mv gitea /usr/local/bin/gitea
```

Maintenant que tous les fichiers/dossiers ont été créés, nous pouvons lancer et installer Gitea. Pour permettre le lancement automatique de Gitea au démarrage de Debian, nous allons créer un **service systemd**.

Pour faciliter la tâche, Gitea fournit sur GitHub un fichier de service systemd déjà prêt. On commence par télécharger le fichier service, qu'on stocke dans le dossier des services systemd:

```
1 sudo wget -O /etc/systemd/system/gitea.service https://raw.githubusercontent.com/go-gitea/gitea/release/v1.21/contrib/systemd/gitea.service
```

Ensuite, on vérifie son contenu, si il correspond avec les configurations faites précédemment.

```
1 sudo nano /etc/systemd/systemd/gitea.service
2 [...]
3 User=git
4 Group=git
5 WorkingDirectory=/var/lib/gitea/
```

Enfin, on lance le service de Gitea avec la commande `systemctl`:

```
1 # Lancer le service Gitea au démarrage du système
2 sudo systemctl enable gitea
3 # Démarrer le service Gitea
4 sudo systemctl start gitea
```

Maintenant que Gitea est lancé, nous pouvons lancer un navigateur, puis se connecter à l'adresse `localhost:3000` pour avoir accès à la page d'installation de Gitea.

- On sélectionne **SQLite3** comme base de données.
- On laisse toutes les autres informations par défaut.



The screenshot shows a web browser window with the address bar set to `localhost:3000`. The page title is "Configuration initiale". A warning message states: "Si vous exécutez Gitea dans Docker, veuillez lire la [documentation](#) avant de modifier les paramètres." The page is divided into two main sections: "Paramètres de la base de données" and "Configuration générale".

Paramètres de la base de données

Gitea nécessite MySQL, PostgreSQL, MSSQL, SQLite3 ou TiDB (avec le protocole MySQL).

Type de base de données *

Emplacement *
Chemin d'accès pour la base de données SQLite3.
Entrer un chemin absolu si vous exécutez Gitea en tant que service.

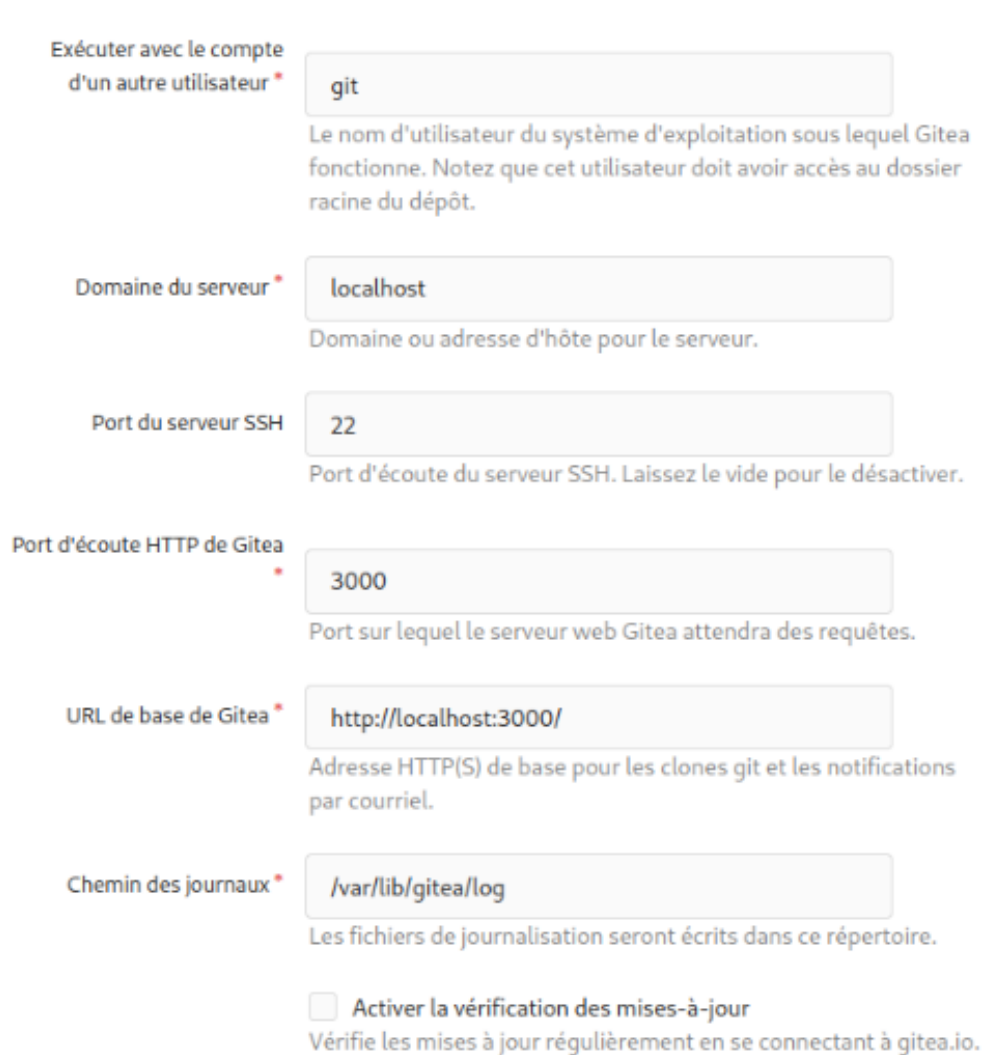
Configuration générale

Titre du site *
Entrez ici le nom de votre société.

Emplacement racine des dépôts *
Les dépôts Git distants seront stockés dans ce répertoire.

Répertoire racine Git LFS
Les fichiers suivis par Git LFS seront stockés dans ce dossier.
Laissez vide pour désactiver LFS.

Figure 9: Configuration de Gitea, part. 1



Exécuter avec le compte d'un autre utilisateur *
Le nom d'utilisateur du système d'exploitation sous lequel Gitea fonctionne. Notez que cet utilisateur doit avoir accès au dossier racine du dépôt.

Domaine du serveur *
Domaine ou adresse d'hôte pour le serveur.

Port du serveur SSH
Port d'écoute du serveur SSH. Laissez le vide pour le désactiver.

Port d'écoute HTTP de Gitea *
Port sur lequel le serveur web Gitea attendra des requêtes.

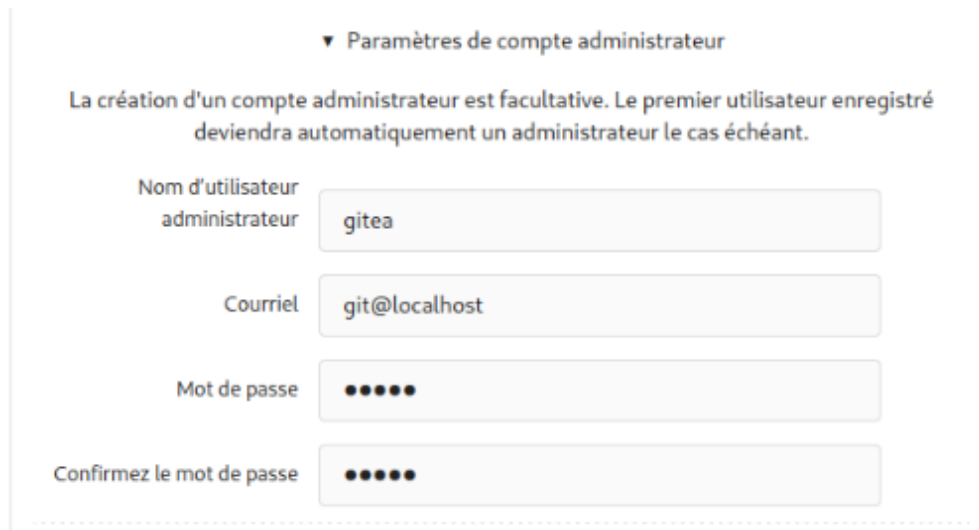
URL de base de Gitea *
Adresse HTTP(S) de base pour les clones git et les notifications par courriel.

Chemin des journaux *
Les fichiers de journalisation seront écrits dans ce répertoire.

☐ Activer la vérification des mises-à-jour
Vérifie les mises à jour régulièrement en se connectant à gitea.io.

Figure 10: Configuration de Gitea, part. 2

- On crée un compte administrateur avec les informations suivantes:
 - **Nom:** *gitea*
 - **Adresse e-mail:** *git@localhost*
 - **Mot de passe:** *gitea*



▼ Paramètres de compte administrateur

La création d'un compte administrateur est facultative. Le premier utilisateur enregistré deviendra automatiquement un administrateur le cas échéant.

Nom d'utilisateur administrateur

Courriel

Mot de passe

Confirmez le mot de passe

Figure 11: Configuration du compte administrateur Gitea

Une fois tout configuré, on peut cliquer sur **Installer Gitea**.

Après une vingtaine de secondes, on arrive sur la page principale de notre interface Gitea.

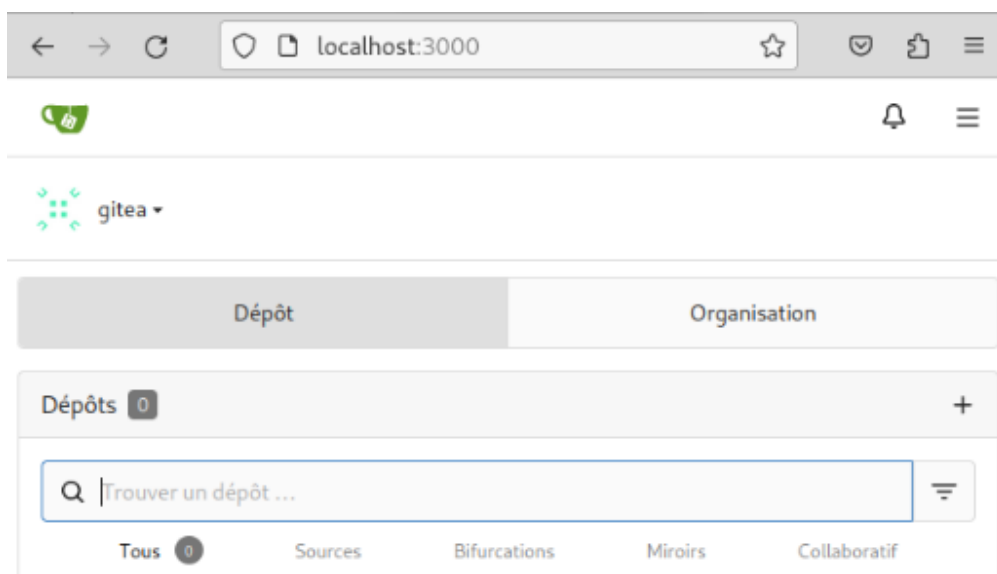


Figure 12: Page principale de Gitea

Une fois l'installation de Gitea finie, nous pouvons protéger les fichiers dans **/etc/gitea**:

```
1 # Arrêt temporaire du service Gitea
2 sudo systemctl stop gitea
```

```
3 # Changement des droits
4 sudo chmod 750 /etc/gitea
5 sudo chmod 640 /etc/gitea/app.ini
6 # Redémarrage de Gitea
7 sudo systemctl start gitea
```

6.4 Mise à jour du binaire de service Gitea

6.4.1 Questions:

Q: Quelle version du binaire avez-vous installé ? Donnez la version et la commande permettant d'obtenir cette information.

A: La version installée est la **1.21.9**. Nous pouvons connaître cette information via la commande suivante:

```
1 gitea --version
```

Q: Comment faire pour mettre à jour le binaire de votre service sans devoir tout reconfigurer ? Essayez en mettant à jour vers la version 1.22-dev.

A: Pour mettre à jour Gitea, il suffit de changer le fichier binaire que nous avons déplacé dans **/usr/local/bin**.

Ici, nous souhaitons mettre à jour de la version **1.21.9** (dernière version stable) à la version **1.22-dev** (dernière version de développement).

Pour commencer, on **arrête le service** Gitea (si il est en cours):

```
1 sudo systemctl stop gitea
```

Ensuite, on télécharge le fichier binaire **nightly** de Gitea:

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

On remplace le précédent fichier binaire de Gitea par le nouveau:

```
1 sudo mv gitea /usr/local/bin/gitea
```

Enfin, on redémarre le service Gitea:

```
1 sudo systemctl start gitea
```

On peut vérifier la version de Gitea avec la commande suivante:

```
1 gitea --version
2 Gitea version 1.22.0+dev-1144-g5c91d7920 [..]
```

6.5 Tests d'utilisation

7 Sources et références

- <https://www.debian.org/>
- https://en.wikipedia.org/wiki/Debian_version_history
- <https://wiki.debian.org/LTS>
- <https://www.debian.org/releases/bullseye/>
- <https://www.debian.org/releases/stable/amd64/apb.en.html>
- <https://www.virtualbox.org/manual/ch04.html>
- <https://git-scm.com/docs/gitk/>
- <https://git-scm.com/docs/git-gui>
- <https://docs.gitea.com/>