

Rapport Installation de Services Réseaux

Version HTML

Clément Franckelemon / Raphaël Piral / Victor Singez

26.03.2023

Table des matières

1	Installation d'une Machine Debian	1
1.1	Installation de l'OS	1
1.1.1	Première étapes à l'installation de l'OS	1
1.1.2	Informations Supplémentaires	1
1.1.3	Installation et configuration de l'OS	2
1.2	Préparation du système	6
1.3	Installation des suppléments invités	6
1.3.1	Installation des suppléments invités depuis le terminal	6
1.3.2	Informations Supplémentaires	7
2	Installation Debian automatisée par préconfiguration	8
2.1	Informations Supplémentaires	8
2.2	Préconfiguration	9
2.3	Installation automatisé	10
3	Installation de Gitea et création de base de données	10
3.1	Configuration Globale de Git	11
3.2	Redirection des ports	11
3.2.1	Accès au port 3000	11
3.3	Installation de Gitea	11
3.3.1	Installation binaire	12
3.4	Utilisation de Gitea depuis le navigateur :	14
3.5	Réalisation de mise à jour	14

1 Installation d'une Machine **Debian**

1.1 Installation de l'OS

1.1.1 Première étapes à l'installation de l'OS

Tous d'abord, nous devons récupérer une ISO bootable pour lancer une installation de Debian 11.

Pour rappel de qu'est-ce qu'un fichier iso bootable : une ISO bootable est un fichier qui **va permettre d'installer des applications**, de **configurer un système**, il contient des composants logiciels qui permettent de **démarrer directement** l'ordinateur sur un menu de configuration/d'installation avant le chargement du système d'exploitation installé sur la machine si il y en a un.

Pour récupérer une ISO Debian, il suffit simplement d'aller sur le site officiel de Debian : <https://www.debian.org/> et de télécharger l'ISO proposée.

Après avoir télécharger l'ISO, on ajoute cette dernière à notre machine VirtualBox en tant que lecteur optique.

1.1.2 Informations Supplémentaires



Table 1: Informations supplémentaire sous forme de foire aux questions

Questions	Réponses
Que signifie “64-bit” dans “Debian 64-bit” ?	Pour information , on dit d’un système “64-bits”, et par extension “Debian 64-bits” signifie un système possédant une architecture 64 bits, soit doublé par rapport à l’architecture d’un “Debian 32-bits”, donc un système 2 fois plus performant qu’un “32-bits” et donc plus adapté à notre usage pour cette SAE, et de façon plus générale, à notre utilisation dans notre vie quotidienne actuelle.(Pour plus d’information sur les système 64 bit : https://www.01net.com/actualites/un-systeme-dexploitation-64-bits-pour-quoi-faire-404210.html)
Quelle est la configuration réseau utilisée par défaut ?	La configuration réseau actuelle sous laquelle nous sommes , est mode d’accès NAT avec un INTEL PRO Desktop (825540EM) comme type d’interface.
Quel est le nom du fichier XML contenant la configuration de votre machine ?	Si vous souhaiteriez un jour réaliser des modification sur la configuration de votre machine virtuelle , il vous suffira d’accéder au fichier de votre Virtual Box et de trouver un fichier XML nommé sae203.vbox-prev .
Comment modifier directement ce fichier pour mettre 2 processeurs à votre machine ?	Par exemple si vous souteriez ajouter un processeur, il vous faudrait ajouter count="2" à la suite de CPU dans l’ouverture de cette balise dans le fichier XML.(exemple ci-joint) <pre><CPU count="2"> <PAE enabled="false"/> <LongMode enabled="true"/> <X2APIC enabled="true"/> <HardwareVirtExLargePages enabled="false"/> </CPU></pre>

1.1.3 Installation et configuration de l’OS

Tout d’abord, on démarre notre machine virtuelle depuis VirtualBox :

Un menu va s’afficher à l’écran, plusieurs choix vont s’offrir à nous mais on va sélectionner l’option “Interface graphique” pour bénéficier d’une configuration simple.

Ensuite, on choisit un nom pour la machine, on va prendre un nom simple “*serveur*”.

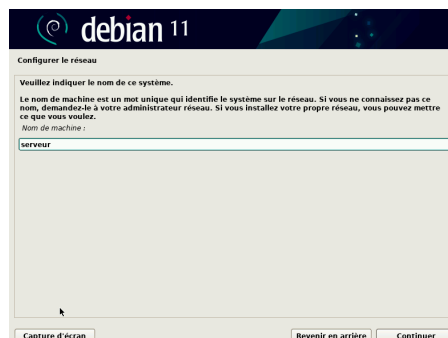


Figure 1: Choix du nom de la machine

Par la suite, on nous propose de choisir un domaine/nom de domaine, mais on peut passer cette étape, elle nous est pas nécessaire.



Figure 2: Choix du nom de domaine

Ensuite on choisit la langue, bien évidemment, nous allons choisir le français.



Figure 3: Choix de la langue

Il nous faut désormais choisir, un miroir¹ Debian pour nous permettre de pouvoir accéder aux différents paquets Debian, images CD-ROM **pour installer, configurer différentes choses sur notre Debian**, mais aussi la consultation des différentes archives des anciennes versions Debian sorties. Nous allons utiliser le miroir le plus proche, celui de Polytech Lille : <http://debian.polytech-lille.fr>.

Continuons avec la configuration du Proxy :

Tout d'abord, rappelons qu'un proxy² permet à un hôte (notre machine par exemple) et un serveur (comme un serveur Web) de **communiquer ensemble** grâce à des échanges de requêtes et réponses HTTP entre les 2 systèmes.

Le proxy que l'on va prendre sera <http://cache.univ-lille.fr:3128>.

¹Pour plus d'information sur les miroir : https://fr.wikipedia.org/wiki/Site_miroir

²Aussi appeler serveur mandataire ([plus d'information sur le serveur mandataire](#))

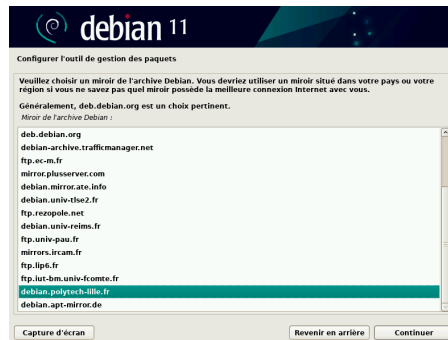


Figure 4: Choix du miroir



Figure 5: Choix du Proxy

Nous nous approchons de la fin mais désormais créons les comptes utilisateurs, commençons par créer le **compte administrateur**, qui possèdera tous les droits possibles sur la machine, on va lui assigner l'identifiant et le mot de passe par défaut qui vont être *root*.

Il faut désormais créer le **compte utilisateur** comme pour le compte administrateur, on va choisir des paramètres basiques, le nom du profil sera *User* et puis l'identifiant et le mot de passe seront donc *user*.

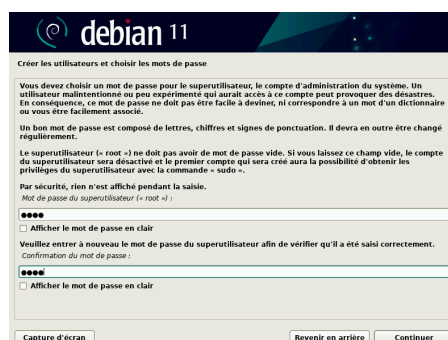


Figure 6: Création du compte root

On s'occupe maintenant de la partition du disque :

On va simplement décider pour réaliser cela de placer tout le contenu de notre système d'exploitation sur une seule partition de disque.



Figure 7: Création du compte User

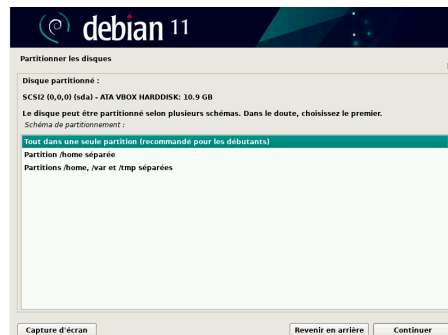


Figure 8: Partitionnage du disque dur

Dernière étape, pour finir l'installation, il nous faut désormais sélectionner les logiciels et paquetages de base de notre Debian, qui vont être l'environnement de bureau Debian, MATE, qui comme GNOME, est un **environnement de bureau**, qui va pouvoir permettre l'utilisation de l'ordinateur via une interface utilisateur graphique. (Pour plus d'information sur l'environnement MATE: <https://fr.wikipedia.org/wiki/MATE>) (Pour plus d'information sur l'environnement GNOME: <https://fr.wikipedia.org/wiki/GNOME>)



Figure 9: Choix des logiciels et paquetages de base

On choisit ensuite de pouvoir utiliser un serveur³ Web⁴ qui pourra nous donner accès à des pages Web, résultats sur Internet, **grâce à des échanges de requêtes et réponses via le protocole HTTP** entre notre machine

³Pour information, il est possible que les serveurs Web et SSH utilisent des serveurs mandataire (proxy) qui sont des serveurs informatiques qui ont pour fonction de relayer des requêtes entre un poste client et un serveur. (pour encore plus d'informations sur ces serveurs mandataires: <https://www.techno-science.net/definition/3812.html>)

⁴Pour plus d'information sur les serveurs Web: https://developer.mozilla.org/fr/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server/

et serveur Web que l'on soit sur un réseau public ou privé.

On a besoin maintenant d'ajouter la possibilité d'utiliser un serveur SSH⁵ qui permet de **recupérer, placer des données** de manière sécurisée **via une demande de connexion auprès du client avec un échange de clés de chiffrement** puis **un encryptage des données** lors du déplacement de ces dernières **via le protocole TCP**.

Puis on sélectionne, pour finir enfin l'installation, quelques utilitaires usuels du système.

1.2 Préparation du système

Pour donner les droit sudo⁶ à un utilisateur il faut , se connecter en temps que super utilisateur. Pour cela appuyer sur **Ctrl + Alt + F1** connecter vous en super utilisateur (login=root et password=root)⁷ puis effectuer la commande suivante⁸ :

```
usermod -aG sudo user
```

Pour en savoir plus sur les groupes auxquels appartient user et s'assurer que user appartient au groupe "sudo" (group des super utilisateur) , Plusieurs cas sont possible pour réaliser cette action :

1. Pour voir tous les groupes existant et les personnes qui y appartiennent il suffit d'utiliser la commande :

```
cat ../../etc/group
```

2. Pour obtenir la liste de tous les groupes dans lequel appartient user , effectuer la commande suivante :

```
groups user
```

(Pour plus d'information sur la commande groups: <https://www.man7.org/linux/man-pages/man1/groups.1.html>)

3. Pour afficher les groupes dans lequel se situe user , on peut utiliser la commande suivante :

```
grep user /etc/group
```

(Pour plus d'information sur la commande grep: <https://www.man7.org/linux/man-pages/man1/grep.1.html>)

1.3 Installation des suppléments invités

Les suppléments invités sont utiliser pour **permettre à une machine virtuelle d'être plus simple d'utilisation et également plus performante**. Elle nous permet, entre-autres, d'adapter la taille de l'écran de la machine virtuelle à la fenêtre occupée par cette dernière, elle permet aussi le partage du presse-papier de la machine virtuelle avec celui de la machine physique.

Pour obtenir les suppléments invités, il va falloir utiliser un terminal pour pouvoir les compiler et rendre utilisable par notre machine virtuelle.

1.3.1 Installation des suppléments invités depuis le terminal

D'abord, dans notre machine virtuelle, dans la barre de menu situé en haut de la fenêtre, sélectionnez *Périphériques* puis *Insérer l'image CD des additions invités...*, ensuite, ouvrez un Terminal depuis le menu *Applications* de la machine virtuelle ou via le raccourci *Ctrl+Alt+T*, entrez la commande :

⁵Pour plus d'information sur les serveur SSH: <https://www.ionos.fr/digitalguide/serveur/outils/protocole-ssh/>

⁶Remarque , si vous ajouter un utilisateur au groupe sudo , alors cette utilisateur devient un super utilisateur.

⁷Si vous avez effectué **l'étape de création de root** comme indiqué précédemment

⁸Pensez bien évidemment à remplacer "user" par le nom d'utilisateur qui vous intéresse et ceux aussi pour les commandes qui suivent.

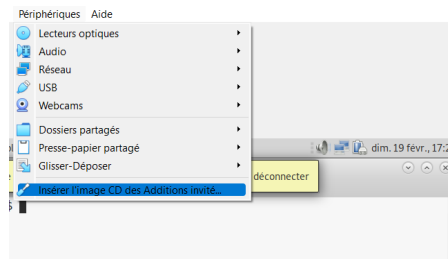


Figure 10: Ajout des suppléments invités

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

La commande `mount`, utilisé dans cette première commande, pour rappel, sert en général à **monter un système** ou à **créer, gérer un système de stockage**, dans notre cas, elle va servir à **positionner le CD supplément invités sur un point de montage** pour y avoir accès. Puis on va saisir la commande suivante qui installera les suppléments sur la machine :

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

Pour vérifier que l'installation à bien été effectuée, on redémarre la machine puis on se connecte avec le compte *user*, on va agrandir la fenêtre ou nous mettre en plein écran pour vérifier que les suppléments ont bien été installés.

Pour éviter des problèmes lors des prochaines configuration à effectuer, on va prendre un "instantané", qui est **une sauvegarde de l'état actuel de la machine**.

1.3.2 Informations Supplémentaires



Quel est la version du noyau Linux utilisé par votre VM ?

Pour connaître la version du noyau utiliser , il faut utiliser la commande :

```
user@machine:~$ uname -r
5.10.0-21-amd64
```

De là on peut constater qu'on utilise la version 5.10.0-21-amd64 , car la commande `uname` permet d'obtenir le nom de la machine , la version noyau et l'architecture , ainsi avec l'option `-r` on obtient uniquement la version du noyau utiliser .

(Pour plus d'information sur la commande `uname` : <https://linuxconfig.org/uname-1-manual-page>)

À quoi servent les suppléments invités ?

Ils permettent de réaliser une configuration plus performante et plus facile d'utilisation , sachant que dans VirtualBox on rencontre des limitation pour le système d'exploitation comme par exemple, vous ne pouvez pas contrôler la résolution de l'écran et vous ne pouvez pas partager le presse-papiers entre un invité et le système d'exploitation hôte. Ainsi les suppléments invités permettent de braver ces limites de par ces suppléments .

(Pour plus d'information sur les supplément d'invité: <https://lecrabeinfo.net/virtualbox-installer-les-additions-invite-guest-additions.html>)

À quoi sert la commande `mount` ?

La commande `Mount` permet de monter un système de fichier , en générale il sert à rattacher un système de fichier à l'arborescence principale . Dans notre cas , la commande a été utiliser afin de se positionner sur un point de montage dans un fichier contenu dans un

périphérique qui est ici directement la VirtualBox , qui sera ainsi rattacher à l'arborescence de fichier de notre système pour être utiliser par la suite.

(Pour plus d'information sur la commande Mount: <https://linux.die.net/man/8/mount>)

2 Installation Debian automatisée par préconfiguration

2.1 Informations Supplémentaires

Voici ci dessous quelques informations interresente sous forme d'une foire aux questions

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

Le “**Porjet Debian**” est une Documentation facile d'accès pour les utilisateurs (Pour plus d'information sur Projet Debian : <https://www.debian.org/doc/>).Le nom Debian provient de la concatenation des noms **Debra**⁹ et **Ian Murdock**¹⁰ qui sont les fondateur du projet (Pour plus d'information sur l'origine du nom Debian : <https://www.debian.org/doc/manuals/debian-faq/basic-defs.fr.html#pronunciation>).

1. 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 ?

Les durées de prise en charge des 3 durée sont :

- la durée **minimale** pour une version Debian correspond à une prise en charge pour **3 ans minimum** pour chaque publication (Pour plus d'information sur la durée minimale d'une version Debian : <https://www.debian.org/releases/index.fr.html>)
- la durée en **support long terme** (LTS) correspond à une durée de vie de **5 ans** (Pour plus d'information sur la durée en support long terme d'une version Debian : <https://www.debian.org/lts/index.fr.html>)
- la durée en **support long terme étendue** (ELTS) correspond à une durée de vie de **10 ans** (Pour plus d'information sur la durée en support long terme étendue d'une version Debian : <https://wiki.debian.org/fr/LTS/Extended>)

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

Les mises à jours de sécurité sont fournie **une année après que la version stable suivante a été publiée , sauf dans le cas où une autre distribution stable est publiée la même année** (Pour plus d'information sur le temps que les mises à jour de sécurité sont fournis : <https://www.debian.org/security/faq.fr.html#lifespan>).

3. Combien de version au minimum sont activement maintenues par Debian ?

Debian a au minimum **3 versions** activement maintenu , parmi elles , une **version “stable”** sous le nom de code Bullseye , une **version “testing”** sous le nom de code Bookworm , une **version “unstable”** sous le nom de code Sid (Pour plus d'information sur le maintient des version actives : <https://www.debian.org/releases/index.fr.html>).

4. D'où viennent les noms de code données aux distributions ?

Les nom de code données aux distribution **provient du responsable du projet Debian** qui travaillait chez Pixar et qui a choisi d'utiliser les **noms provenant de Toy Story** (Pour plus d'informations sur la provenance des noms de code Debian : <https://www.debian.org/doc/manuals/debian-faq/ftparchives.fr.html#sourceforcodenames>).

5. L'un des atouts de Debian fut le nombre d'architecture officiellement prises en charge. Combien et lesquelles sont prises en charge par la version Bullseye ?

Debian prend en charge 9 architectures principale et de nombreuses variantes avec Debian 11 : AMD64 & Intel 64 , Intel x86-based , ARM , ARM avec matériel FPU , ARM 64 bits , MIPS 64 bits (petit-boutien) , MIPS 32 bits (petit-boutien) , Power Systems , IBM S/390 64 bits . (Pour lus d'information sur les architectures pris en charge par Debian : <https://www.debian.org/releases/stable/arm64/ch02s01.fr.html>)

⁹Il s'agit du nom de la femme du créateur du projet

¹⁰Pour plus d'information sur Ian Murdock : https://fr.wikipedia.org/wiki/Ian_Murdock

6. Première version avec un nom de code

- Quelle a été le premier nom de code utilisé ?

Le premier nom de code utilisé fut **buzz** .

- Quand a-t-il été annoncé ?

Cette version est apparue le **26 avril 2004** .

- Quelle était le numéro de version de cette distribution ?

Il s'agit de la **version Debian 1.1** .

Pour plus d'information sur le 1er nom de code et sa date d'apparition : <https://www.debian.org/doc/manuals/debian-faq/ftparchives.fr.html#sourceforcodenames>

Pour plus d'information sur le numéro de la 1er version à avoir eu un nom de code : https://www.debian.org/social_contract.1.1

7. Dernier nom de code attribué

- Quelle a été le premier nom de code utilisé ?

Le dernier nom de code annoncée à ce jour est **Forxy** .

- Quand a-t-il été annoncé ?

Il a été annoncé le **8/11/20** .

- Quelle était le numéro de version de cette distribution ?

La version de cette distribution est **Debian 14** .

Pour plus d'information sur le dernier nom de code et la version correspondante : <https://wiki.debian.org/DebianReleases>

Pour plus d'information sur la date d'annonce de Debian 14 : <https://wiki.debian.org/DebianTrixie>

2.2 Préconfiguration

Pour réaliser une installation automatisée , pour cela nous aurons besoin de récupérer et préparer les fichiers nécessaires à notre automatisation . Pour commencer nous devons récupérer le fichier **autoinstall.zip** , décompressez le , et puis insérez le dans le répertoire de la machine. Avec les fichiers que l'on vient de récupérer , nous allons pouvoir avoir une préconfiguration minimale , à laquelle nous allons ajouter les morceaux de configurations que l'on souhaite avoir à l'installation. Ainsi nous aurons principalement 2 fichiers à modifier : **preseed.cfg** et **vboxpostinstall.sh** .

Parmi les deux fichiers cités précédemment , **preseed.cfg**¹¹ nous permettra de réaliser la configuration que l'on pouvait retrouver dans une installation manuelle , sous forme de liste d'ensemble d'informations et actions automatisées pour l'installation . Tandis que **vboxpostinstall.sh** est un fichier qui va plutôt nous permettre de réaliser la configuration de notre bureau suite à l'installation du système d'exploitation.

Ainsi pour commencer nous allons réaliser la préconfiguration du système d'exploitation , et ceux en allant dans le fichier **preseed.cfg** . De là nous pourrions modifier par exemple , l'utilisateur standard et le super utilisateur . Ainsi si l'on veut par exemple **ajouter les droits de super utilisateur à l'utilisateur standard** , nous pouvons ajouter **“sudo”** dans le groupe de l'utilisateur comme ci-dessous (**les groupes de l'utilisateur standard sont ici dans la dernière ligne**) :

```
## Utilisateur standard
d-i passwd/user-fullname string User
d-i passwd/username string user
d-i passwd/user-password password user
d-i passwd/user-password-again password user
d-i passwd/user-default-groups string audio cdrom video sudo
```

¹¹Pour plus de détails sur les diverses possibilités de préconfiguration : <https://www.debian.org/releases/stable/amd64/apb.fr.html>

Ensuite si nous souhaitons par exemple **avoir un environnement de bureau** et non juste un terminal , nous pouvons ajouté comme dans l'exemple qui suit , **l'environnement MATE** en ajouté **"mate-desktop"** comme dans la dernière ligne :

```
## Installation meta-paquetages
# Tâches à installer (via des méta-paquetages)
# Lister les possibilités : tasksel --list-task (en ligne de commande)
# Utiliser au minimum "standard" est une bonne idée
tasksel tasksel/first multiselect standard ssh-server,mate-desktop
```

Une fois notre Système d'exploitation installer , nous pouvons passe à la suite en installant à présent tout ce dont on à besoin dans notre bureau. et pour cela on va à passer dans le fichier **vboxpostinstall.sh**¹².

Ainsi si l'on souhaite par exemple **ajouter des paquets** , il vous suffit d'entrer la commande que vous auriez entrer dans un terminal depuis votre bureau mais avec écrit devant **"log_command_in_target"** , dans le fichier **vboxpostinstall.sh** comme suit :

```
#
# Packages needed for GAs.
#
echo "-----" >> "${MY_LOGFILE}"
echo '** Installing packages for building kernel modules...' | tee -a "${MY_LOGFILE}"
log_command_in_target apt-get -y install build-essential
log_command_in_target apt-get -y install linux-headers-${uname -r}
log_command_in_target apt-get install --yes git-all
log_command_in_target apt-get install --yes sqlite3
log_command_in_target apt-get install --yes curl
log_command_in_target apt-get install --yes bash-completion
log_command_in_target apt-get install --yes neofetch
```

Ainsi dans l'exemple ci dessus , on installe de la **ligne 8 à 12** , les paquets suivant :

- git
- sqlite3
- curl
- bash
- neofetch

2.3 Installation automatisé

Maintenant que la préconfiguration est terminer , nous pouvons à présent passer à l'installation avec les fichiers préconfigurer au par avant. Pour commencer il va nous falloir **modifier identifiant unique universel**¹³ du fichier **S203-Debian11.viso** et ceux en se plaçant dans le répertoire de la machine virtuel que l'on à crée au par avant , puis d'exécuter la commande suivante :

```
sed -i -E "s/(--iprt-iso-maker-file-marker-bourne-sh).*$/\1=$(cat /proc/sys/kernel/random/uuid)/" S203-Debian11.viso
```

Ensuite, il ne vous suffit plus qu'à **inserez le fichier S203-Debian11.viso dans le lecteur** de la machine, et à lancez votre machine qui va ainsi s'installer automatiquement à son démarrage.

3 Installation de Gitea et création de base de données

Tout d'abord, pour pouvoir installer Gitea, il vous faut un debian, en version 11.6 minimum, avec *mate-desktop*, *git* puis *sqlite3* et les autres paquets qui vous ont été demander d'installer dans l'étape précédente : [*Installation Debian automatisée par pré-configuration*].

¹²Pour plus de détails sur ce qu'est ce fichier : <https://connect.ed-diamond.com/GNU-Linux-Magazine/glmfhs-045/une-installation-de-debian-automatique>

¹³Pour plus d'informations sur les identifiant unique universel : https://fr.wikipedia.org/wiki/Universally_unique_identifier

3.1 Configuration Globale de Git

Après vous êtes connectez avec votre compte utilisateur et ouvert un terminal, il vous suffit d'entrer dans votre terminal les commandes `git config --global user.name "Prénom Nom"`, `git config --global user.email "votre@email"` avec comme "user" votre nom utilisateur, et `git config --global init.defaultBranch "master"`, cette étape vous permettra d'éviter des problèmes lors de la création de votre branche par défaut de votre git de projet.

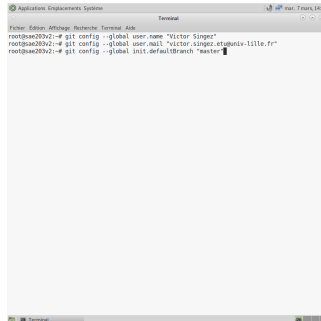


Figure 11: Configuration Git

Ensuite, installer le paquet Git-Gui depuis le terminal, avec la commande `apt-get install git-gui`.

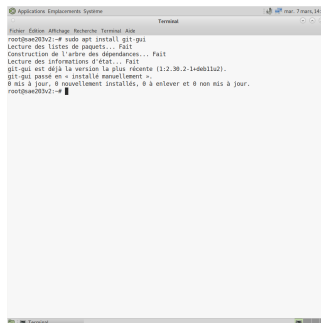


Figure 12: Installation Git Gui

Git-Gui est un git avec une interface graphique, il se lance avec la commande suivante : `git gui [] []`. Pour en savoir plus : <https://git-scm.com/docs/git-gui>.

GitK est un git fonctionnant sous forme de navigateur de dépôt git, il se lance en utilisant la commande : `gitk [] [] [-] []`, plus d'information sur GITK : <https://git-scm.com/docs/gitk>.

Pour utiliser par défaut le proxy de votre université par exemple, sur l'ensemble de vos projets Git, il faudra utiliser la commande Git : `git config --global http.proxy http://[proxyuser]:[proxypwd]@[proxy.server.com]:[8080]` avec pour *proxyuser* votre nom d'utilisateur, *proxypwd* votre mot de passe et *proxy.server.com* l'URL de votre serveur, pour plus d'informations, consultez le lien suivant : https://itsmycode.com/configure-git-proxy/?utm_content=cmp-true.

3.2 Redirection des ports

3.2.1 Accès au port 3000

Pour pouvoir utiliser Gitea, il vous faut utiliser le port 3000 sur votre machine physique et virtuelle comme dans notre cas. Plus d'informations ici : <https://www.virtualbox.org/manual/UserManual.html#networkingdetails>.

Pour cela, dans *VirtualBox*, allez dans le menu de configuration de votre machine virtuelle, puis dans la catégorie *Réseau* cliquez sur *Redirection de ports* puis sur *Ajouter une nouvelle règle de redirection* et réalisez la configuration suivante :

3.3 Installation de Gitea

Tout d'abord, Gitea est une forge logicielle web écrite en Golang qui permet de fournir une interface web de divers dépôts git gérer par une ou plusieurs personnes, c'est un logiciel open source. Plus d'informations : <https://fr.wikipedia.org/wiki/Gitea>.

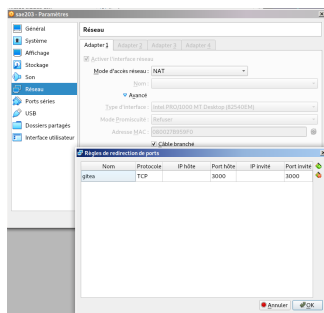


Figure 13: Redirection des Ports

On peut comparer Gitea à Gitlab mais aussi Github qui sont aussi des dépôts Git.

3.3.1 Installation binaire



Attention : Pour l'installation, vous devez, pour certaines étapes, être connecté en mode *root* dans votre terminal, pour cela entrez la commande *su root* et entrez le mot de passe de votre compte *root*. Cela sera précisé quand ce sera le cas.

-Etape 1 (*root*) : Téléchargez les fichiers binaires de Gitea avec *wget* comme ceci :

```
user@debian:~$ wget https://dl.gitea.com/gitea/1.18.5/gitea-1.18.5-linux-amd64
--2023-08-15 10:10:10-- https://dl.gitea.com/gitea/1.18.5/gitea-1.18.5-linux-amd64
Resolving dl.gitea.com (dl.gitea.com)... 157.140.10.10
Connecting to dl.gitea.com (dl.gitea.com):80: connected.
HTTP/1.1 200 OK
Saving to: 'gitea-1.18.5-linux-amd64'
gitea-1.18.5-linux-amd64 100% |#####| 1.1 MB 0:00:01
user@debian:~$
```

Figure 14: Téléchargements des données binaires

-Etape 2 : (*root*) : Faites la même chose que précédemment pour le fichier *.asc* :

```
user@debian:~$ wget https://dl.gitea.com/gitea/1.18.5/gitea-1.18.5-linux-amd64.asc
--2023-08-15 10:10:10-- https://dl.gitea.com/gitea/1.18.5/gitea-1.18.5-linux-amd64.asc
Resolving dl.gitea.com (dl.gitea.com)... 157.140.10.10
Connecting to dl.gitea.com (dl.gitea.com):80: connected.
HTTP/1.1 200 OK
Saving to: 'gitea-1.18.5-linux-amd64.asc'
gitea-1.18.5-linux-amd64.asc 100% |#####| 1.1 MB 0:00:01
user@debian:~$
```

Figure 15: Téléchargements des données binaires

Etape 3 : (*root*) : Vérifier les signatures GPG des fichiers récupérés :

Etape 4 : Il est recommandé pour le bon fonctionnement du serveur Web, la commande suivante :

Etape 5 : Avant de continuer il est conseillé de vérifier que vous avez bien une version 2.0 de Git minimum avec la commande suivante :

Etape 6 (*root*) : Créer un utilisateur pouvant utiliser Gitea comme ceci :

Etape 7 (*root*) : Créer les fichiers nécessaires pour former la structure de Gitea :

Etape 8 (*root*) : Donnez-vous les droits sur les fichiers Gitea et *app.ini* pour y avoir accès :

Etape 9 (*root*) : Exportez le fichier Gitea et déplacez le fichier binaire :

Les étapes suivantes vont vous permettre d'avoir un démarrage automatique du service Gitea.

Etape 10 (*root*) : Déplacez-vous dans le fichier *system* puis créez et complétez le fichier *gitea.service* (récupérez le code ici : <https://github.com/go-gitea/gitea/blob/main/contrib/systemd/gitea.service>).

Etape 11 (*root*) : Démarrez ensuite le service Gitea :

Etape 12 (*root*) : Installez ensuite *supervisor* :

Etape 13 (*root*) : Créez le fichier permettant l'utilisation de *supervisor* et déplacez-vous y. **Note :** créez ce fichier à l'endroit où est installé votre Gitea.

Etape 14 (*root*) : Créez et complétez le fichier *supervisord.conf* (récupérez le code ici : <https://github.com/go-gitea/gitea/blob/main/contrib/supervisor/gitea>).

```

user@debian:~$ git --verify gitea.asc gitea

```

Figure 16: Vérifications des signatures

```

root@debian:/home/user# GITEA_CUSTOM=/home/gitea/custom ./gitea web

```

Figure 17: Recommandation configuration serveur

```

user@debian:~$ git --version
git version 2.38.2

```

Figure 18: Vérification version de Git

```

user@debian:~$ sudo adduser \
--system \
--shell /bin/bash \
--gecos 'Git Version Control' \
--group \
--disabled-password \
--home /home/git \
git

```

Figure 19: Création de l'utilisateur

```

user@debian:~$ su root
root@debian:/home/user# mkdir -p /var/lib/gitea/{custom,data,log}
chown -R git:git /var/lib/gitea/
chown -R 750 /var/lib/gitea/
mkdir /etc/gitea
chown root:git /etc/gitea
chown 770 /etc/gitea

```

Figure 20: Création des structures pour Gitea

```

root@debian:/home/user# chown 750 /etc/gitea
chown 640 /etc/gitea/app.ini

```

Figure 21: Vérifications des signatures

```

root@debian:/home/user# export GITEA_WORK_DIR=/var/lib/gitea/

```

Figure 22: Exportez fichiers de Gitea

```

root@debian:/home/user# cp gitea /usr/local/bin/gitea

```

Figure 23: Déplacez fichiers binaire de Gitea

```

root@debian:/home/user# cd /etc/systemd/system
root@debian:/etc/systemd/system# nano gitea.service

```

Figure 24: Créer le fichier gitea.service

```

cat < gitea.service
[Unit]
Description=Gitea (Git with a cup of tea)
After=syslog.target
After=network.target
# Don't forget to add the database service dependencies
#
# Requires=mysql.service
# After=mysql.service
# Requires=redis.service
# After=redis.service
# Requires=memcached.service
# After=memcached.service
#
# Requires=redis.service
# After=redis.service
#
[Service]
Type=simple
User=git
Group=git
WorkingDirectory=/var/lib/gitea
ExecStart=/usr/local/bin/gitea web

```

Figure 25: Compléter le fichier gitea.service

```

root@debian:/etc/systemd/system# sudo systemctl enable gitea
sudo systemctl start gitea

```

Figure 26: Démarrage de gitea.service

```

root@debian:/etc/systemd/system# sudo apt install supervisor

```

Figure 27: Installation de supervisor

```

root@debian:/etc/systemd/system# mkdir /home/git/gitea/log/supervisor

```

Figure 28: Création fichier pour supervisor

```

root@debian:/etc/systemd/system# cd /etc/supervisor/

```

Figure 29: Déplacement dans fichier supervisor

```

root@debian:/etc/supervisor# nano supervisord.conf

```

Figure 30: Créer le fichier supervisord.conf

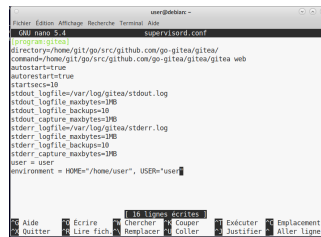


Figure 31: Compléter le fichier supervisor.conf



Figure 32: Démarrage supervisor

Etape 15 (root) : Démarrez ensuite *supervisor* :

Etape 16 : Vérifier, pour finir, le bon fonctionnement de Gitea :



Figure 33: Vérification fonctionnement de Gitea

3.4 Utilisation de Gitea depuis le navigateur :

Lancer le navigateur de votre machine physique et ouvrez l'URL : <http://localhost:3000/> :

Pour pouvoir utiliser la version Web de Gitea, choisissez en type de base de données *SQLite3* et réalisez les faits suivants :

3.5 Réalisation de mise à jour

Mettre à jour Gitea est possible, pour cela arrêter le fonctionnement de Gitea (avec *root*):

Modifier ensuite la valeur de version de Gitea et récupérez une version plus récente de Gitea via *wget* (avec *root*):

Puis, redémarrez Gitea (avec *root*) et vérifiez son bon fonctionnement :

If you run Gitea inside Docker, please read the [documentation](#) before changing any settings.

Database Settings

Gitea requires MySQL, PostgreSQL, MSSQL, SQLite3 or TiDB (MySQL protocol).

Database Type: **MySQL**

Host: **127.0.0.1:3306**

Username: **gitea**

Password:

Database Name: **gitea**

Note: MySQL users please use the InnoDB storage engine and if you use SQLite3, your SQLite3 version must be greater than 3.8

Charset: **utf8mb4**

General Settings

Site Title: **Gitea: Git with a cup of tea**

You can enter your company name here.

Repository Root Path: **/var/lib/gitea/data/repo**

Remove Git repositories will be used to this directory.

Figure 34: Utilisation de Gitea depuis le Web

Database Settings

Gitea requires MySQL, PostgreSQL, MSSQL, SQLite3 or TiDB (MySQL protocol).

Database Type: **SQLite3**

Figure 35: Choix type de base de données

Administrator Account Settings

Creating an administrator account is optional. The first registered user will automatically become an administrator.

Administrator Username: **gitea**

Password: *********

Confirm Password: *********

Email Address: **gitea@localhost**

Install Gitea

Figure 36: Paramétrage serveur Web de Gitea

```
root@debian:~# systemctl stop gitea
```

Figure 37: Arrêt fonctionnement de Gitea

```
root@debian:~# curl https://dl.gitea.com/gitea-1.19-linux-amd64
```

Figure 38: Modification version de Gitea

```
root@debian:~# mv /tmp/gitea /usr/local/bin
root@debian:~# chmod +x /usr/local/bin/gitea
```

Figure 39: Récupération version plus moderne

```
root@debian:~# systemctl restart gitea
```

Figure 40: Redémarrage Gitea

```
root@debian:~# systemctl status gitea.service
gitea.service - Gitea (Git with a cup of tea)
Loaded: loaded (/etc/systemd/system/gitea.service; vendor preset: enabled)
Active: active (running) since Sat 2023-03-25 15:40:53 CET; 11s ago
Main PID: 3048 (gitea)
Tasks: 8 (limit: 2322)
Memory: 1.0M
CGroup: /system.slice/gitea.service
└─3048 /usr/local/bin/gitea web --config /etc/gitea/app.ini
```

Figure 41: Vérification fonctionnement Gitea