

Installer un serveur Turnkey sur Raspberry PI

Table of Contents

- [Installer un serveur Turnkey sur Raspberry PI](#)
 - [Installation et configuration de Raspbian](#)
 - [Installer Raspbian](#)
 - [Configurer Raspbian](#)
 - [Installation de Turnkey LXC](#)
 - [Installer Turnkey Core](#)
 - [Problèmes les plus courants](#)
 - [Installer l'appliance Turnkey LXC](#)

Cet article détaille la mise en place et la configuration d'un serveur **Turnkey** comprenant toutes les fonctionnalités standard de:

- **TurnKey Core**: le système d'exploitation de base que toutes les solutions **TurnKey** GNU/Linux partagent. Il est généralement déployé de manière autonome comme point de départ pratique pour les intégrations système personnalisées. Les avantages incluent des mises à jour de sécurité quotidiennes automatiques, une sauvegarde et une restauration en un clic, un panneau de configuration Web et une surveillance du système préconfigurée avec des alertes par e-mail en option.
- **TurnKey LXC**: simplifie le téléchargement et le déploiement de plusieurs applications **TurnKey** côte à côte sur le même hôte dans des conteneurs légers isolés de manière sécurisée tout en gérant des détails délicats tels que le routage réseau. **LXC** (AKA Linux Containers) est la technologie de virtualisation légère et montante qui alimente Docker et d'autres plates-formes de déploiement de logiciels de nouvelle génération.

Installation et configuration de Raspbian

Installer Raspbian

Télécharger l'image raspbios buster armhf lite

```
wget  
https://downloads.raspberrypi.org/raspbian_lite_armhf/images/raspbian_lite_armhf-2021-05-28/2021-05-07-raspbian-buster-armhf-lite.zip]]
```

Copier l'image Buster sur la carte SD, en utilisant l'une des méthodes décrites [ici](#).

Insérer la carte sd et allumer le PI

Se connecter via la console en utilisant le login par défaut : pi/raspberry Ouvrir le menu "raspi-config":

1. Etendre le système de fichier de la carte sd
2. Définir les paramètres régionaux, le fuseau horaire
3. dans le menu "AVANCE", mettre le gpu ram à 16mb
4. dans le menu "AVANCE", Activer ssh

Redémarrer le PI

```
sudo reboot
```

Maintenant on a un PI HEADLESS. cela signifie qu'on a plus besoin de clavier ou de moniteur, connecter simplement le port ethernet du pi, on utilisera SSH pour continuer la suite du tutoriel.



Deskpi Pro est un kit matériel pour convertir une Raspberry PI 4 standard à partir d'un SBC nu, avec un stockage limité, en un mini PC complet avec un bouton d'alimentation, un refroidissement, de meilleurs ports et un Mass Storage via SATA USB3 ou M.2. Pour l'installer sur Raspbian 64bit il faut installer git en premier `apt-get update && apt-get -y install git` puis procéder ainsi:

```
cd ~
git clone https://github.com/DeskPi-Team/deskpi.git
cd ~/deskpi/
./install.sh
```

Configurer Raspbian

Trouver l'adresse IP PI.

Se connecter en ssh depuis un terminal:

```
$ ssh pi@"VOTRE_PI_IP"
```

Entrez le mot de passe par défaut "raspberry"

Définir un mot de passe ROOT :

```
$ sudo passwd root
```

Se déconnecter du pi :

```
exit
```

Se reconnecter à root :

```
$ ssh root@"VOTRE_IP_PI"
```

Le Raspbian est prêt maintenant.

Installation de Turnkey LXC

Installer Turnkey Core

Raspliance Core est un remix de Raspian/Raspbian qui vise à tirer le meilleur parti des potentiels de Raspberry Pi en tant que plate-forme serveur.

C'est autant un remix de Raspian/Raspbian que de TurnKey Linux Core 12.0, dont il s'inspire.

La plupart du travail est fait avec le script

<https://github.com/ghoulmann/Raspliance-Core/blob/master/core/conf>:

- Purge tous les packages suivants : scratch, lxde, xserver-xorg, lightdm, lightdm-gtk-greeter, gnome-icon-theme, gnome-themes-standard, consolekit, gtk2-engines, lxsession
- Installe les paquets suivants: adduser, apt, base-files, base-passwd, bash, bash-completion, bind9-host, bsdmainutils, bsdutils, ca-certificates, console-terminus, coreutils, cpio, cron, cron-apt, dash, debconf, debconf-i18n, debian-archive-keyring, debianutils, dialog, diffutils, dmsetup, dpkg, e2fslibs, e2fsprogs, eject, etckeeper, ethtool, file, findutils, gcc-4.4-base, gettext-base, git, git-core, gnupg, gpgv, grep, groff-base, gzip, hdparm, hostname, ifupdown, insserv, iproute, iptables, iputils-ping, kbd, keyboard-configuration, klibc-utils, less, libacl1, libattr1, libauthen-pam-perl, libblkid1, libbsd0, libbz2-1.0, libc-bin, libc6, libcap2, libcomerr2, libcurl3-gnutls, libdebconfclient0, libdebian-installer4, libdevmapper1.02.1, libedit2, liberror-perl, libexpat1, libfontconfig1, libgcc1, libgcrypt11, libgdbm3, libgeoip1, libgnutls26, libgpg-error0, libgssapi-krb5-2, libidn11, libio-pty-perl, libk5crypto3, libkeyutils1, libklibc, libkrb5-3, libkrb5support0, libldap-2.4-2, liblocale-gettext-perl, libmagic1, libncurses5, libncursesw5, libnet-ssleay-perl, libnewt0.52, libnfnetwork0, libopts25, libpam-modules, libpam-runtime, libpam0g, libparted, libpopt0, libreadline5, libreadline6, librsync1, libsasl2-2, libsasl2-modules, libsepol1, libslang2, libsqlite3-0, libss2, libstdc++6, libtasn1-3, libtext-charwidth-perl, libtext-iconv-perl, libtext-wrapi18n-perl, libudev0, libusb-0.1-4, libuuid-perl, libuuid1, libwrap0, libxml2, localepurge, locales, login, logrotate, lsb-base, lsb-release, lvm2, man-db, mawk, mime-support, mktemp, module-init-tools, mount, nano, ncurses-base, ncurses-bin, net-tools, netbase, ntp, ntpdate, openssh-blacklist, openssh-client, openssh-server, openssl, os-prober, passwd, perl, perl-base, perl-modules, procs, psmisc, python, python-central, python-crypto, python-minimal, python-pycurl, python-simplejson, python-support, python2.6, python2.6-minimal, readline-common, resolvconf, rsync, rsyslog, screen, sed, sensible-utils, shellinabox, ssh, syslinux-common, sysv-rc, sysvinit, sysvinit-utils, tar, traceroute, tzdata, ucf, udev, udhcpc, unzip, util-linux, uuid-runtime, vim-common, vim-tiny, wget, whiptail, xkb-data, xz-utils, zlib1g, libbind9-80, libdns81, libgmp3-dev, libisc83, libisccc80, libiscfg82, liblwres80, liblzma5, libssl1.0.0





- Installe <http://code.turnkeylinux.org/pythondialog/>
- Installe <http://code.turnkeylinux.org/pycurl-wrapper/>
- Installe <http://code.turnkeylinux.org/turnkey-pylib/>
- Construit /etc/confconsole.conf à partir de <http://code.turnkeylinux.org/inithooks/debian/init>
- Active les services webmin (Webmin et ses modules: écoutent sur le port 12321), confconsole, inithooks, ssh, shellinabox defaults
- Configure shellinabox (ajax web shell) pour qu'il écoute sur le port 12320 `sed -i "s|4200|12320|" /etc/default/shellinabox`

Se connecter en ssh sur ROOT:

```
$ ssh root@"VOTRE_IP_PI"
```

Cloner le repo git dans le PI :

```
$ git clone https://github.com/ghoulmann/Raspliance-Core /tmp/tkl
```

Aller dans /tmp/tkl et compiler le patch tkl :

Suite à la réorganisation du repo code.turnkeylinux.org, remplacer dans le fichier core/conf:

1. les `wget http://code.turnkeylinux.org/pythondialog/` en `wget http://code.turnkeylinux.org/pythondialog/tree/master`
2. les `http://code.turnkeylinux.org/turnkey-pylib/` en `wget http://code.turnkeylinux.org/turnkey-pylib/tree/master`
3. les `wget http://code.turnkeylinux.org/pycurl-wrapper/` en `wget http://code.turnkeylinux.org/pycurl-wrapper/tree/master/`
4. `wget http://code.turnkeylinux.org/inithooks/debian/init` par `wget http://code.turnkeylinux.org/inithooks/tree/master/debian/init hooks.init\`



ou

appliquer le patch suivant sur le fichier core/conf:

```
<codedoc toggle conf.diff> — conf 2021-11-19 09:11:49.524143381 +0000 +++ conf
2021-11-19 09:16:22.112139430 +0000 @@ -240,9 +240,9 @@ #Install Python-Dialog
mkdir /tmp/pythondialog cd /tmp/pythondialog/ -wget
http://code.turnkeylinux.org/pythondialog/setup.py -wget
http://code.turnkeylinux.org/pythondialog/dialog.py -wget
http://code.turnkeylinux.org/pythondialog/setup.cfg +wget
http://code.turnkeylinux.org/pythondialog/tree/master/setup.py +wget
http://code.turnkeylinux.org/pythondialog/tree/master/dialog.py +wget
http://code.turnkeylinux.org/pythondialog/tree/master/setup.cfg #Setup PythonDialog
python setup.py install
```

```
@@ -250,49 +250,49 @@ mkdir -p /tmp/pycurl-wrapper/pylib mkdir -p /tmp/pycurl-  
wrapper/debian cd /tmp/pycurl-wrapper/debian -wget  
http://code.turnkeylinux.org/pycurl-wrapper/debian/control +wget  
http://code.turnkeylinux.org/pycurl-wrapper/tree/master/debian/control cd /tmp/pycurl-  
wrapper -wget http://code.turnkeylinux.org/pycurl-wrapper/setup.py +wget  
http://code.turnkeylinux.org/pycurl-wrapper/tree/master/setup.py cd pylib/ -wget  
http://code.turnkeylinux.org/pycurl-wrapper/pylib/pycurl\_wrapper.py +wget  
http://code.turnkeylinux.org/pycurl-wrapper/tree/master/pylib/pycurl\_wrapper.py cd  
/tmp/pycurl-wrapper/ python setup.py install
```

```
#Install Turnkey Pylib mkdir -p /tmp/turnkey-pylib/ cd /tmp/turnkey-pylib/ -wget  
http://code.turnkeylinux.org/turnkey-pylib/setup.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/setup.py mkdir -p /tmp/turnkey-  
pylib/pylib/ cd pylib -wget http://code.turnkeylinux.org/turnkey-pylib/pylib/command.py -  
wget http://code.turnkeylinux.org/turnkey-pylib/pylib/conffile.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/debian\_pylib.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/executil.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/fifobuffer.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/fileevent.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/lazyclass.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/multiprocessing\_utils.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/netinfo.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/paths.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/pidlock.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/popen4.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/signore.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/stdtrap.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/sysversion.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/temp.py -wget  
http://code.turnkeylinux.org/turnkey-pylib/pylib/threadloop.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/command.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/conffile.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/debian\_pylib.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/executil.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/fifobuffer.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/fileevent.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/lazyclass.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/multiprocessing\_utils.py  
+wget http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/netinfo.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/paths.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/pidlock.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/popen4.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/signore.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/stdtrap.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/sysversion.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/temp.py +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/pylib/threadloop.py cd  
/tmp/turnkey-pylib mkdir debian cd debian/ -wget  
http://code.turnkeylinux.org/turnkey-pylib/debian/control +wget  
http://code.turnkeylinux.org/turnkey-pylib/tree/master/debian/control cd /tmp/turnkey-  
pylib python setup.py install cd ~/
```





```
#Inithooks conf cd /etc -wget http://code.turnkeylinux.org/inithooks/debian/init -mv  
/etc/init /etc/confconsole.conf +wget  
http://code.turnkeylinux.org/inithooks/tree/master/debian/inithooks.init +mv  
/etc/inithooks.init /etc/confconsole.conf  
  
#Update rc.d for services update-rc.d webmin defaults </codedoc>
```

```
$ cd /tmp/tkl/  
$ ./compile_tklpatch.sh
```

Installer TurnKey dans le chemin système :

```
$ tklpatch-apply / ./core/
```

attendre que tout soit fini pour redémarrer le pi :

```
$ shutdown -r now
```

ou

```
$ reboot
```

Connecter à nouveau le clavier et le moniteur au PI

Au premier démarrage, il faut entrer le mot de passe root (choisir le mot de passe le plus puissant possible).

Problèmes les plus courants

On peut rencontrer des problèmes pendant ou après l'installation.

Impossible de se connecter avec shellinabox

Shellinabox n'est pas un client ssh, c'est juste une application qui émule le shell d'un système distant via un navigateur Web. En cas d'impossibilité de se connecter à shellinabox, se connecter en ssh puis éditer /etc/hosts et changer 127.0.1.1 de "core" en "mynewname"

UNCONFIGURED INTERFACES

- Solution 1 : désactiver la dépendance

Quitter la confconsole. se connecter en root, éditer le fichier /etc/network/interfaces pour

ajouter la ligne suivante en haut du fichier :

```
# UNCONFIGURED INTERFACES
```

Ou exécuter cette commande :

```
sed -i '1i# UNCONFIGURED INTERFACES' /etc/network/interfaces
```

- Solution 2 : Désactiver le service confconsole

(UNIQUEMENT si le correctif ne fonctionne pas) quitter la confconsole. se connecter à root puis exécuter:

```
/etc/init.d/confconsole stop  
update-rc.d -f confconsole remove
```

Installer l'appliance Turnkey LXC

Installer les paquets prérequis: lxc, dnsmasq, dnsmasq-utils, bridge-utils, nginx, apt-cacher-ng, python-requests, ebttables, xz-utils, aptitude, python-apt, python-lxc, python-httpplib2, bzip2

Lorsque TurnKey LXC est déployé sur bare-metal, la configuration la plus simple est d'utiliser un pont, auquel cas le conteneur se voit attribuer une adresse IP sur le réseau (via DHCP) et tous ses services (par exemple, SSH, serveur Web, etc.) sont accessibles directement.

Ajouter un périphérique réseau de pont nommé br0 en créant un fichier à l'aide de la commande suivante, avec le contenu ci-dessous :

```
sudo vi /etc/systemd/network/bridge-br0.netdev
```

```
[NetDev]  
Name=br0  
Kind=bridge
```

Ajouter l'interface Ethernet intégrée (eth0) en tant que membre du pont en créant le fichier suivant :

```
sudo vi /etc/systemd/network/br0-member-eth0.network
```

```
[Match]  
Name=eth0  
  
[Network]  
Bridge=br0
```



Le logiciel du point d'accès ajoutera l'interface sans fil wlan0 au pont au démarrage du service. Il n'est pas nécessaire de créer un fichier pour cette interface. Cette situation est particulière aux interfaces LAN sans fil.

Activer maintenant le service systemd-networkd pour créer et remplir le pont lorsque le Raspberry Pi démarre :

```
sudo systemctl enable systemd-networkd
```

dhcpcd, le client DHCP sur le Raspberry Pi, demande automatiquement une adresse IP pour chaque interface active. il faut donc bloquer le traitement des interfaces eth0 et wlan0 et laisser dhcpcd configurer uniquement br0 via DHCP.

```
sudo vi /etc/dhcpcd.conf
```

Ajouter la ligne suivante près du début du fichier (au-dessus de la première ligne xxx de l'interface, le cas échéant) :

```
denyinterfaces wlan0 eth0
```

Aller à la fin du fichier et ajoutez ce qui suit :

```
interface br0
```

Avec cette ligne, l'interface br0 sera configurée conformément aux valeurs par défaut via DHCP. Enregistrer le fichier pour terminer la configuration IP de la machine.

Cloner le repo de l'appliance Turnkey LXC

```
git clone https://www.turnkeylinux.org/lxc
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-turnkey>

Last update: **2025/02/19 10:59**

