

# Cluster de Raspberry Pi





## Remerciements

Nous remercions Madame Camille Cotti pour ses conseils et ses directions au sein de ce projet qui fut attrayant et enrichissant de toutes parts. Nous profitons, par la même occasion, de remercier l'établissement pour nous avoir fourni le matériel adéquate et nécessaire afin de réaliser pleinement ce projet.



# Table des matières

1. Explication du sujet
  - a) But du projet
  - b) Liste des équipements et services
2. Mise en place d'une image minimale
  - a) Image système
  - b) Minimisation de l'image et stockage sur la carte SD
  - c) Premier pas et configuration de Raspbian
3. Logiciel PXE
  - a) Présentation du logiciel
  - b) Etape de démarrage par le PXE
  - c) Installation du logiciel PXE

## Introduction

De nos jours, nous retrouvons des machines dans toutes les entreprises, parfois très nombreuses, ce qui augmente la difficulté de configurations de celles-ci. En effet, un technicien peut se retrouver à gérer une centaines d'ordinateurs au sein de son entreprise, et lorsqu'il devra mettre à jour celles-ci, par exemple, il lui faudra énormément de temps.

Or, ceci est évitable aujourd'hui car il existe l'**automatisation**. Mais comment peut-on définir ce terme qui est à la fois clair et obscur?

L'automatisation, que l'on évoquera tout au long de notre projet, est un dispositif, qui reproduit en autonomie une séquence d'actions prédéterminées sans que l'Homme n'intervienne. Par conséquent, si le technicien évoqué précédemment utilise ce processus, il n'aura qu'à intervenir sur une seule machine, celle du serveur par exemple puis les autres machines viendront récolter les mises à jour sur le serveur.

Pour notre projet, nous avons à notre disposition 8 Raspberry Pi, dont nous détaillerons les composants au fil du rapport, mais aussi des cartes SD de 8go chacune, ainsi que les matériaux périphériques qui nous permettront notamment le branchement des Raspberry Pi.

Ce que nous avons essayé de reproduire avec ces 8 Raspberry Pi est l'architecture de plusieurs ordinateurs auxquels nous tenterons d'automatiser certaines tâches. Ce sera pour nous une première expérience que l'on peut d'ores et déjà qualifier de professionnel.

Par conséquent, nous allons suivre un plan à 3 parties. Tout d'abord nous détaillerons le projet en lui même ainsi que ces enjeux, puis nous évoquerons la mise en place de l'image minimale pour les Raspberry, et enfin nous expliquerons l'utilisation d'un logiciel PXE.

## I. Explication du sujet

### (a) But du projet

L'intitulé de notre projet est le suivant : « Cluster de Raspberry Pi ». Peu sont ceux qui comprendront directement le sens de ce titre. Commençons donc par définir le terme « cluster ». Un cluster, qui, littéralement signifie « grappe de calcul », désigne un ensemble de machines rassemblées pour effectuer des calculs, souvent lourds.

Les Raspberry-Pi mis à notre disposition sont tous identiques. En effet, l'établissement nous a fourni ces machines vierges, c'est à dire sans aucune configuration au préalable, ainsi, nous aurons pour mission d'installer un système d'exploitation sur chacune d'entre elles. Ce système d'exploitation sera le plus minimale possible, seul un terminale sera nécessaire. En effet, notre but sera d'automatiser le déploiement du véritable système d'exploitation ainsi que sa configuration sur toutes les machines fournies. Pour cela, nous avons le choix entre 2 possibilités :

- Un script, qui aura pour but d'aller chercher, au démarrage de la machine, le système d'exploitation complet présent sur le serveur. Après rédaction du script, il sera mis dans la carte SD du Raspberry.
- L'autre choix est l'automatisation par le réseau. En effet, il est possible de déployer le système d'exploitation ou d'effectuer des mises à jour en passant par un logiciel PXE (Preboot Execution Environment). Celui ci permet de booter depuis le réseau en récupérant l'image sur le serveur.

Nous avons choisis d'utiliser le logiciel boot PXE dont nous allons évoquer les qualités ainsi que le fonctionnement dans la partie III.

Ce projet nous permettra d'acquérir de réelles sensations vis à vis du monde professionnel. En effet, nous serons ici confrontés à une situation qui se reproduira quoiqu'il arrive en entreprise, car, comme nous le savons tous, l'automatisation est un procédé particulièrement riche et sera de plus en plus utilisé à l'avenir.

Les Raspberry sont capables de communiquer entre eux via le réseau et donc s'échanger des informations. Nous devons donc nous appuyer sur ce système de communication afin de réaliser les objectifs de notre projet. Nous verrons donc dans la partie suivante les différents équipements et services qui nous permettent par exemple de communiquer.

## (b) Liste des équipement et services

La majorité des équipements ont été mis à notre disposition par l'université. Allons voir de plus près leurs composants utiles à la réalisation de notre projet.

- Les 8 Raspberry Pi sont les équipements nécessaires à notre projet. En effet, c'est par leur biais que nous allons pouvoir créer une situation qui pourrait se dérouler en entreprise. Un Raspberry Pi est doté d'une prise HDMI, ce qui nous permettra notamment de le brancher à un écran afin de visualiser nos manipulations. Ses 2 ports USB et son port carte SD nous permettront respectivement de brancher un clavier, une souris et une carte SD qui contiendra notre système d'exploitation minimal. On dispose de surcroît, d'un port Ethernet. Les Raspberry seront connectés à un switch qui se chargera de fournir le réseau Ethernet à nos 8 machines. Ce type de réseau est le plus répandu en entreprise. En effet quasiment toutes les machines vendues actuellement possèdent des cartes de mise en réseau de type

Ethernet. Il se distingue car il est assez rapide, avec des débits allant de 10 Méga Bits par secondes à 10 Giga Bits par secondes.

→ D'autre part, nous disposerons d'une machine serveur qui détiendra le véritable système d'exploitation ; c'est cette image que chaque Raspberry Pi devra venir cueillir. La machine serveur n'est autre qu'un simple ordinateur, nous utiliserons un PC quelconque avec une configuration Linux. De plus, nous utiliserons un serveur DHCP qui délivrera des adresses IP aux Raspberry localisé sur le réseau de notre machine serveur. Enfin, un serveur TFTP sera nécessaire car celui-ci nous permettra d'envoyer des fichiers sur le réseau, d'une machine à une autre, dans notre cas ce sera l'image du système qui sera transférée. La machine serveur sera connectée au switch qui lui sera connecté au 8 Raspberry comme nous l'avons indiqué précédemment; ils seront donc sur le même réseau.

### (c) Intérêt du projet

Ce projet est pour nous une première expérience professionnelle car nous serons confrontés à une situation réelle, qui s'effectue tous les jours dans bon nombre d'entreprises qui détiennent plusieurs ordinateurs. La situation en entreprise est bien entendu beaucoup plus complexe, mais le principe y est. En effet, on retrouve en général des centaines de PC qui doivent être automatisés. Notre projet, quant à lui sera effectué avec 8 Raspberry Pi. Évidemment le principe reste le même et nous pouvons donc ajouter ces nouvelles compétences à notre CV.

Nous avons acquis des notions solides sur l'automatisation, et nous avons mis en pratique nos connaissances tel que TFTP pour les transferts de fichiers.



## II. Mise en place d'une image minimale

### (a) Image système

L'image du système d'exploitation a été facilement téléchargeable car la marque Raspberry est mondialement connue. Par conséquent, après quelques recherches nous sommes tombés sur un système d'exploitation qui se nomme Raspbian. Nous l'avons rapidement choisie du fait de sa légère taille ainsi que ses compétences. Raspbian est en fait, la fusion du mot « Raspberry » et « Debian ». C'est un système d'exploitation libre, basé sur Debian, que nous connaissons très bien. C'est donc, comme son nom l'indique, une modification de Debian pour les équipements légers tels que les Raspberry Pi. Nous avons téléchargé l'image à l'adresse suivante <https://www.raspberrypi.org/downloads/raspbian/> et ensuite, nous l'avons décompressée.

Pourquoi avoir choisi Raspbian ? Comme nous le savons, les Raspberry Pi sont des ordinateurs avec des ressources limitées. Ainsi, Raspbian qui contient des logiciels légers, nous permet de remplir parfaitement le cahier des charges. Par exemple, l'environnement de bureau de Raspbian est « Pixel » ; c'est un environnement léger et amélioré pour Raspberry Pi.

Dans notre projet, le véritable système d'exploitation sera celui que le Raspberry Pi viendra cueillir sur la machine serveur à l'aide du logiciel boot PXE. Mais comment allons-nous configurer un Raspberry Pi qui a priori n'a pas de système d'exploitation ? Pour être clair, le Raspberry aura un système d'exploitation dans la carte SD de celui-ci, qui sera minimisée afin simplement de pouvoir réaliser nos commandes utiles à l'automatisation.

## (b) Minimisation de l'image et stockage sur la carte SD

L'image Raspbian qui a été téléchargé sur le site <https://www.raspberrypi.org/downloads/raspbian/> est « Raspbian Jessie Lite » qui est plus légère car elle ne contient pas d'environnement de bureau. C'est celle-ci qui sera injecté dans la carte SD du Raspberry Pi.

Quant à l'autre système d'exploitation « Raspbian Jessie With Pixel », c'est celle-ci qui sera sur notre serveur et qui devra être transférée sur le Raspberry Pi. Comme son nom l'indique, elle contient le fameux « Pixel » évoqué ci-dessus.

Après avoir récupéré Raspbian Jessie Lite, nous avons installer l'image disque sur la carte SD à l'aide du logiciel Win32DiskImager. Cela nous a pris une dizaines de minutes.

Ensuite nous avons inséré la carte SD dans notre Raspberry, puis nous avons branché notre machine à l'écran via le port HDMI et pour finir nous avons branché un clavier une souris et un câble Ethernet sur notre Raspberry.

## (c) Premier pas et configuration de Raspbian

Premièrement, nous avons mis notre système en langue française comme ceci :

Choisir l'option **"4. Internationalisation Options"**  
Trois sous-menu apparaissent que nous allons tous utiliser :

### → **Change**

### **Locale**

— Choisir les le mode (avec la barre ESPACE du clavier)  
"fr\_FR.UTF-8 UTF-8"

— Cocher ou laisser coché le mode en\_GB.UTF-8 UTF-8

- Valider les choix (touche TAB et bouton OK)
- Par défaut, choisir fr\_FR.UTF-8"
- **Change** **Timezone**
  - Choisir "Europe"
  - Choisir Paris
- **Change** **Keyboard** **Layout**
  - Choisir "Generic 105-key (Intl PC)
  - Choisir ensuite "Other"
  - choisir ensuite "French"

Deuxièmement, nous avons testé la connexion à distance via SSH. Pour ce faire, nous allons décrire les étapes pour aboutir à ce mode de connexion :

- Choisir l'option "**8 Advanced Options**"
  - Choisir "A4 SSH"

Maintenant il est possible de se connecter en SSH, ce qui est un point positif car nous n'aurons plus à transporter le clavier et la souris ainsi que notre grand écran, tout sera sur notre ordinateur auquel le Raspberry sera connecté.



### III. Logiciel PXE

#### (a) Présentation du logiciel

PXE signifie Pre-boot Execution Environment. C'est un logiciel qui va permettre à nos Raspberry de démarrer depuis le réseau en récupérant une image de système d'exploitation qui se trouve sur notre serveur. En d'autre terme c'est un environnement de démarrage d'ordinateurs en réseau.

L'image qu'il récupère est notre système d'exploitation personnalisé avec des composantes logicielles tel que bureautique, utilitaires, packs de sécurité, scripts, etc... Ainsi cette image « pré-chargée » sera installée sur nos machines qui ont été amorcées en PXE, c'est-à-dire sur nos Raspberry.

#### (b) Etape de démarrage par PXE

- Il faut tout d'abord le configurer dans le BIOS qui se trouve fréquemment dans un menu concernant la carte réseau. Le BIOS (Basic Input Output System) est un ensemble de fonctions, contenu dans la mémoire morte (ROM) de la carte mère de notre serveur, lui permettant d'effectuer des opérations de base lors de sa mise sous tension, par exemple l'identification des périphériques d'entrée/sortie connecté (nos Raspberry) et la demande aux cartes réseaux d'activer leur ROM PXE.
- La ROM PXE va par la suite demander au serveur DHCP de notre réseau de lui fournir une adresse IP. Le serveur DHCP (Dynamic Host Configuration Protocol) est un protocole réseau qui assure la configuration automatique des paramètres TCP/IP de nos Raspberry

en lui fournissant notamment une adresse IP ainsi qu'un masque de sous réseau.

- Par la même occasion le serveur DHCP va donner à la ROM PXE le fichier à télécharger par TFTP et le serveur sur lequel ils se connecteront. TFTP (Trivial File Transfert Protocol) est un protocole simplifié de transfert de fichiers.
- La ROM PXE va envoyer certaines informations au serveur PXE comme par exemple l'ID de la carte réseau et va ensuite télécharger le fichier indiqué par le serveur DHCP via TFTP.
- La machine envoie une requête TFTP pour charger pxelinux.0 au serveur TFTP. Une fois le fichier pxelinux.0 récupéré, la machine l'exécute. Celui-ci va alors récupérer la configuration PXE auprès du serveur.

La taille du fichier à amorcer ne permet pas de « booter » directement un noyau Linux mais il faut que le logiciel à amorcer le télécharge et l'exécute lui-même. C'est alors que le fichier récupéré auprès du TFTP va continuer le téléchargement du système (par exemple PXELinux.0), c'est-à-dire qu'il va télécharger des fichiers qui seront mis en RAM sur chaque Raspberry et va monter un partage réseau SAMBA qui est un protocole permettant à des systèmes Unix d'accéder aux ressources de ces systèmes.

A la fin de ces différentes étapes donc, lors de la mise sous tension de nos 8 Raspberry, nous pourrions constater que le logiciel PXE va bien chercher le système d'exploitation personnalisé sur notre serveur afin de pouvoir l'utiliser sur nos Raspberry. On rappelle que cela nous permet d'avoir automatiquement les mises à jour ainsi que les modifications de notre système d'exploitation personnalisé, faite sur le serveur, sur nos 8 Raspberry en même temps. Cela permet donc de gagner du temps.

## (c) Installation du logiciel PXE

Installation des services sur le serveur :

- Serveur DHCP :
  - Apt-get install dhcp3-server
  - On le configure ensuite dans le fichier /etc/dhcp3/dhcp.conf avec un éditeur de texte comme gedit par exemple.
  - On devra par la suite vérifier si ce serveur à bien attribué à chaque Raspberry une adresse IP personnelle.
  
- Serveur SAMBA :
  - Apt-get install samba
  - On le configure ensuite dans le fichier /etc/samba/smb.conf avec un éditeur de texte comme gedit par exemple.
  - Après la configuration on relance le service : /etc/init.d/samba restart
  
- Serveur TFTP :
  - Apt-get install tftpd-hpa
  - Installation :
    - Création de l'arborescence :
      - > mkdir /tftpboot
      - > chown nobody /tftpboot/ #Pour que les Raspberry puissent accéder au TFTP
      - > su nobody
      - > mkdir /mnt/Images/Boot
      - > mkdir /mnt/Images/Boot/reminst/ #Répertoire qui va contenir les binaire de démarrage du Windows PE
      - > chmod 777 /mnt/Images/Boot/ -R #Droits de lecture, écriture, exécution
      - > exit

- > ln -s /mnt/Images/Boot/tftpboot #Raccourci d'un lien symbolique
- > su nobody
- > chmod 777 /tftpboot/ -R
- Configuration des paramètres de démarrage : gedit /etc/default/tftpd-hpa
- Après la configuration on lance le service : /etc/init.d/tftpd-hpa start
- Création du client TFTP : > apt-get install tftp
  - > Su nobody
  - > Gedit tftpboot/test.txt
  - > Tftp @IP

### Mise en place du PXE :

- Installation de Syslinux :
  - Apt-get install syslinux
  - Cp /usr/lib/syslinux/pxelinux.0 /mnt/Images/Boot
- Configuration de PXELinux : Le fichier « pxelinux.0 » que nous venons de copier est un binaire. Il sert à charger d'autre binaire compatible ou charger un noyau linux avec son image virtuelle.
  - Su nobody
  - Mkdir /mnt/Images/Boot/pxelinux.cfg
  - Gedit /mnt/Images/Boot/pxelinux.cfg/default
- Mise en place des fichiers de démarrage PXE de Windows PE : c'est un système d'exploitation exécuté le temps de la sauvegarde sur le client. La fondation du système qu'on peut appeler le « noyau » même sera en mémoire vive du client (RAM) puis le partage des réseaux va charger les compléments annexes par le biais du protocole SAMBA.

Les fichiers utilisés viennent du CD-ROM d'installation de Windows Server 2003 :

- Décompresser les fichiers : `expand -r CD-ROM:\I386\fichier.??_`
  - `CD \D Z:\boot`
  - `COPY CD-ROM:\I386\SETUPLDR.BIN .\NTLDR`
  - `COPY CD-ROM:\I386\NTDETECT.COM`
  - `COPY CD-ROM:\I386\startrom.n12 .\xpe.0`
- Installation de sys-linux : Le système de démarrage PXE va charger Windows depuis le réseau. Pour cela il va passer par un service PXE qui contient BINL (Boot Information Network Layer): le client envoie au serveur PXE des informations sur sa carte réseau, et le serveur va regarder dans sa base de connaissance s'il a un driver adapté pour la carte réseau du client.
- `Wget http://oss.netfarm.it/guides/ris-linux-0.2.tar.gz`
  - `Tar zxvf ris-linux-0.2.tar.gz`
  - `Cd ris-linux-0.2`
  - `Mkdir /tftpboot/patch/`
  - `Mv * tftpboot/patch/`
  - `Cd tftpboot/patch/`
  - `Chmod 777 *`
- Configuration et exécution du script : Le script va en fait renommer les fichiers de l'image de base Windows PE pour correspondre à la bonne casse. Il a une variable « REP » qui indique où se trouve l'image Windows PE.
- `Gedit ./fixup-repository.sh`
  - `> REP=/tftpboot/reminst/XPE`
  - `./fixup-repository.sh`



- `Chmod 777 /tftpboot/reminst/XPE/ -R`
  
- Création de base de données : Nous devons créer une base de données des cartes réseaux supportée par l'image de Windows PE basique afin que notre client démarre dessus. Ce script écrit en Python permet de « scanner » les fichiers INF qui correspondent aux drivers réseau. Pour créer cette base de données exécutons le script avec comme paramètre le dossier qui contient les fichiers INF.
  - `Apt-get install python`
  - `./infparser.py /tftpboot/XPE/i368/inf`
  
- Test de l'image de base de Windows PE : Une fois le serveur redémarré, on teste voir si notre image de base fonctionne. Pour cela, on branche un Raspberry sur le switch, puis on le démarre sur le réseau (toucher « F12 » pour y accéder).

## Conclusion

Notre projet tuteuré qui c'est donc étalé sur toute l'année et mis en place par notre tutrice Madame Camille Coti a eu pour objectif final donc de créer un réseau totalement « automatisé » à partir d'un serveur, d'un switch et de 8 Raspberry, pour différents critères cité notamment dans notre première partie.

On a donc ajouté sur nos 8 Raspberry une image minimale afin de les configurer, une image personnalisé sur notre serveur et un logiciel PXE qui va jouer l'intermédiaire entre les différents équipements de notre réseau afin de récupérer sur nos Raspberry l'image personnalisé de manière automatique, d'obtenir par la même occasion les mises à jour et les modifications faites sur notre image personnalisé directement sur nos Raspberry lors de leur mise sous tension, sans agir manuellement déçus.

Pour conclure, ce projet nous a donc permis d'expérimenter un nouvel équipement informatique nommé « Raspberry », d'un nouveau logiciel appelé PXE, d'utiliser directement des systèmes d'exploitations choisis, créés et modifiés par nous-même et de réutiliser des services, des commandes, des manières de travailler utilisés dans le cadre de notre DUT tel que le serveur DHCP, l'équipement Switch ou encore un éditeur de texte Gedit.

Notre projet fût très enrichissant et très captivant dans le cadre de notre DUT.