

Rapport d'Analyse : Déploiement Automatisé d'un Parc Informatique Linux

Ethan Rogge, Bui The Dat, Melnic Alexandru, Flament Franklin, Bollengier Tho

15 mai 2025

1 Introduction

L'entreprise ouvre un nouveau bureau avec 100 postes de travail sous Linux pour ses développeurs. Actuellement, l'installation et la configuration des machines sont réalisées manuellement, ce qui est inefficace et sujet à des erreurs. Ce rapport vise à proposer une solution automatisée garantissant un déploiement rapide, sécurisé et homogène des postes de travail, tout en tenant compte des différences matérielles et des besoins spécifiques des utilisateurs.

2 Problème à Résoudre

Les principales difficultés du déploiement manuel sont :

- **Temps de déploiement trop long** : Installation manuelle prenant plusieurs jours.
- **Manque d'homogénéité** : Risques de différences entre postes.
- **Mises à jour et maintenance complexes** : Impossibilité d'appliquer des modifications globales rapidement.
- **Hétérogénéité des machines** : Tous les postes de travail n'ont pas la même configuration matérielle, ce qui nécessite une adaptation des logiciels et des réglages en fonction des performances disponibles.

Une solution automatisée est nécessaire pour garantir une configuration cohérente et un gain de temps considérable, tout en s'adaptant aux spécificités matérielles des postes.

3 Choix des Solutions Techniques

3.1 Automatisation de l'Installation

1. Autoinstall (solution retenue) :

- Répond automatiquement aux questions de l'installateur Ubuntu.
- Compatible avec PXE/iPXE pour une installation réseau sans intervention humaine.
- Facilement personnalisable selon les profils et configurations matérielles.

2. Alternatives :

- **Kickstart** : Adapté aux distributions RedHat, donc non pertinent pour Ubuntu.
- **Preseed** : Plus ancien est incompatible avec les version 24+ de ubuntu
- **FAI (Fully Automated Install)** : Très flexible mais plus complexe à mettre en œuvre.

Conclusion : Preseed est retenu pour sa simplicité et sa compatibilité avec PXE/iPXE.

3.2 Automatisation de la Maintenance

1. Ansible (solution retenue) :

- Aucune installation d'agent sur les machines (contrairement à d'autres solutions).
- Gestion flexible des configurations selon les profils utilisateurs et les performances matérielles.
- Permet des mises à jour et des modifications à grande échelle.

2. Alternatives :

- **Puppet** : Plus adapté aux très grandes infrastructures mais nécessite un agent.
- **Chef** : Complexe à apprendre et basé sur Ruby.
- **SaltStack** : Très puissant, mais plus lourd à mettre en place.
- **Terraform** : Plus orienté Cloud, donc moins adapté pour un déploiement local.

Conclusion : Ansible est retenu pour sa facilité d'utilisation et son efficacité dans un parc informatique hétérogène.

4 Infrastructure Technique et Serveurs Nécessaires

4.1 Serveur PXE/iPXE pour le Déploiement Réseau

Pour automatiser l'installation à grande échelle, un serveur PXE/iPXE est indispensable. Il permet aux machines de démarrer via le réseau et de télécharger l'image d'installation sans nécessiter de support physique (USB, DVD). Ce serveur joue un rôle central dans le processus d'amorçage des machines.

Composants du serveur PXE/iPXE :

- **Serveur DHCP** : Assigne une adresse IP aux machines et indique l'emplacement du serveur PXE.
- **Serveur TFTP** : Fournit l'image de boot PXE/iPXE et les fichiers de configuration.
- **Serveur HTTP** : Utilisé pour le chargement rapide des fichiers d'installation avec iPXE.
- **Serveur d'installation** : Héberge les fichiers autoinstall pour une installation automatisée et pour la post-configuration.

Processus de démarrage PXE/iPXE :

1. Le client démarre sur le réseau et demande une IP au serveur DHCP.
2. Le serveur DHCP répond avec l'adresse du serveur PXE/iPXE.
3. Le client télécharge et exécute le bootloader via TFTP (ou HTTP avec iPXE).
4. Le chargeur PXE/iPXE télécharge l'installateur Ubuntu et le fichier Preseed, puis lance l'installation automatique.
5. Une fois Ubuntu installé, autoinstall est utilisé pour appliquer les configurations spécifiques aux profils des machines.

Cette approche permet un déploiement centralisé et rapide, tout en garantissant une homogénéité des configurations.

5 Mise en œuvre et planning

5.1 Serveurs nécessaires

- **Serveur de déploiement** : Contient DHCP, PXE, TFTP et autoinstall pour gérer l'installation et la configuration automatique des postes.

5.2 Phases du projet

1. **Installation et configuration des serveurs**
2. **Définition et test des profils de configuration**
3. **Automatisation du déploiement**
4. **Validation et ajustements**
5. **Déploiement global**
6. **Documentation et formation des équipes**

5.3 Profils et Personnalisation des Configurations

Ansible permet de gérer plusieurs profils en fonction des ressources matérielles et des rôles des utilisateurs :

- **Développeurs** : Docker, Python, Git, c++, Java, VsCode, IntelliJ, Node.js (dernier version stable).
- **Administrateurs** : Virtual box, Wireshark, Apache, FAI.
- **Secretaires et manageur** : Libre Office.
- **Tous les utilisateurs** : Outils communs (Zoom, OpenVPN, UFW, ClamAV, Outlook).

5.4 Bénéfices attendus

- **Gain de temps** : Installation et configuration des machines en quelques heures au lieu de plusieurs jours.
- **Adaptabilité** : Chaque machine est configurée selon ses besoins spécifiques.
- **Maintenance simplifiée** : Les mises à jour et installations sont centralisées.
- **Sécurité accrue** : Toutes les machines suivent des configurations validées et homogènes.
- **Réduction des coûts** : Moins de techniciens mobilisés et diminution des erreurs humaines.

6 Conclusion

Grâce à PXE/iPXE, Autoinstall Ansible, l'entreprise bénéficie d'un déploiement automatisé, rapide et sécurisé, adapté aux spécificités matérielles

et aux besoins de chaque développeur. Cette approche garantit une gestion simplifiée du parc informatique, réduisant les interventions manuelles et optimisant les performances des postes de travail.