

# Déploiement des postes clients

Sylvain Allemand, Jacquelin Charbonnel, Damien Ferney, Benoit Métrot

Rencontres Mathrice de printemps - Mai 2025



# Plan

Faddef

Diskfull à l'IMB

Postes fixes au LAREMA

Kit Ubuntu

Synthèse



# FaDDeF

FaDDeF<sup>1</sup> est un système de déploiement rapide et simplifié de systèmes GNU/Linux sur un ensemble de PC hétérogènes sans disque dur (diskless ubuntu) c'est à dire un poste qui :

- ▶ démarre sur le réseau
- ▶ n'a pas besoin de DD mais peut en avoir un (swap, fichiers locaux....)
- ▶ exploite les ressources locales.



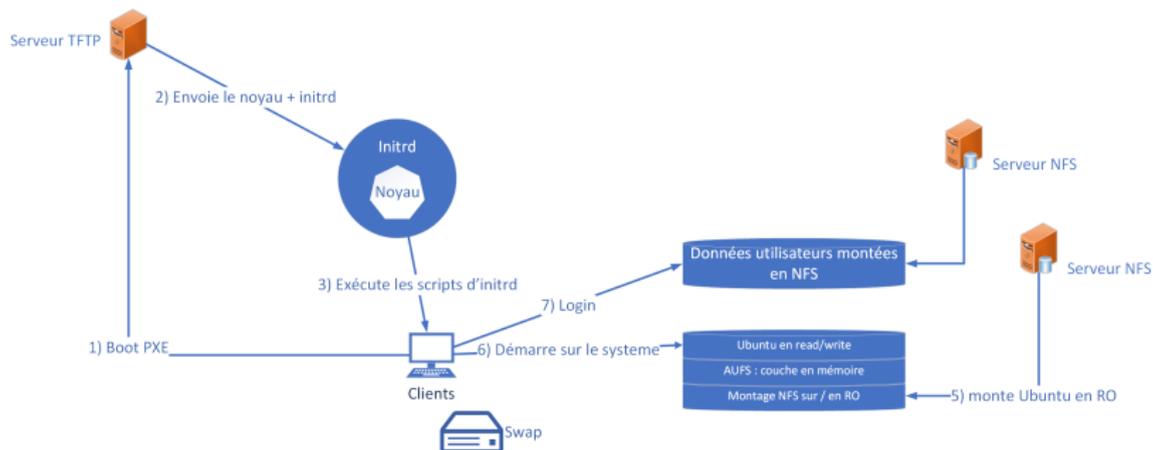
---

<sup>1</sup>Philippe Depouilly, Zouhir Hafidi. Déploiement simplifié de stations sans disque avec Faddef. JRES (Journées réseaux de l'enseignement et de la recherche ) 2009, Renater, Dec 2009, Nantes, France. hal-04804290

# Besoins

- ▶ Gestion centralisée (ajout logiciels, mise à jour, modification de configuration → une opération en seul point)
- ▶ Ensemble de logiciels communs à tous les postes
- ▶ Poste opérationnel dès le déballage
- ▶ Menu de démarrage pour sélection de la distribution / version

# Fonctionnement



# Fonctionnement

- ▶ Allumage de l'ordinateur et boot PXE.
- ▶ Configuration IP par DHCP
- ▶ Récupération du noyau et de l'image initrd par TFTP
- ▶ Montage du système racine en ro par NFS avec par dessus un espace RW en mémoire (unionfs, aufs)
- ▶ Lancement des services
- ▶ Ouverture de session par l'utilisateur
- ▶ fichiers utilisateurs sur montage NFS (rw, nfsv4)

Les applications s'exécutent sur le CPU et la mémoire de l'ordinateur à partir des programmes et fichiers de configurations disponibles sur le montage NFS. Toutes les machines partagent le même système racine.



# Construction du système de fichier exporté par NFS

- ▶ A partir d'un poste maître configuré aux petits oignons (ou bootstrap + chroot):
  - ▶ identification et authentification
  - ▶ services d'impression (éventuellement locaux)
  - ▶ accès aux home
  - ▶ paquets et logiciels
  - ▶ log distants, ...
- ▶ Script de création du système NFS tricks et hacks spécifiques à chaque distribution (pas de copie de /dev, /proc, /var/log, ....)
- ▶ Script <sup>2</sup> de construction et compilation de l'image initrd spécialisée et du noyau associé (pour distribution par TFTP à l'aide de mkinitramfs ou update-initramfs)

# Construction du système de fichier exporté par NFS

- ▶ `initramfs-tools` :
  - ▶ construit un `initramfs` bootable qui sera lu avec le noyau (`vmlinuz`) et sera responsable de monter le root filesystem et de démarrer l'init du système.
  - ▶ fichier de conf et arborescence de scripts dans `/etc/initramfs-tools/` (`BOOT=nfs`, `MODULES=netboot`, drivers noyau nécessaires au démarrage : réseau, aufs, `resolv.conf`, `/etc/hostname`, `/etc/hosts`, clavier, ....)
  - ▶ `/etc/initramfs-tools/scripts` contient les scripts lancés aux différents instants du démarrage :
    - *init-top* les premiers,
    - *init-bottom* les derniers, *avant* ou *après* montage root nfs, ...



Ne passe pas à Ubuntu supérieur à 20.04 à cause de la disparition du module noyau AUFS.

# Installation / Remplacement d'une nouvelle machine

1. Déballer et brancher la machine
2. Insérer une section MAC-address + NOM + IP dans le serveur DHCP
3. Compléter le serveur DNS (ou dynamique)
4. Allumer la machine et configurer le boot PXE systématique



# Emplacement des données utilisateurs

- ▶ Montage du répertoire d'accueil par NFS
- ▶ Utilisation en mode kiosk (clef usb possible)



# Limitations

1. très dépendant d'un bon réseau
2. poste forcément fixe même si données utilisateurs ne sont plus en NFS.
3. ubuntu  $\geq$  22.04 aufs obsolete, et overlayfs incompatible avec / sur NFS .
4. pas très compatible avec systèmes actuels (snap, .local, .cache firefox, etc ...)
5. trouver d'autres solutions...

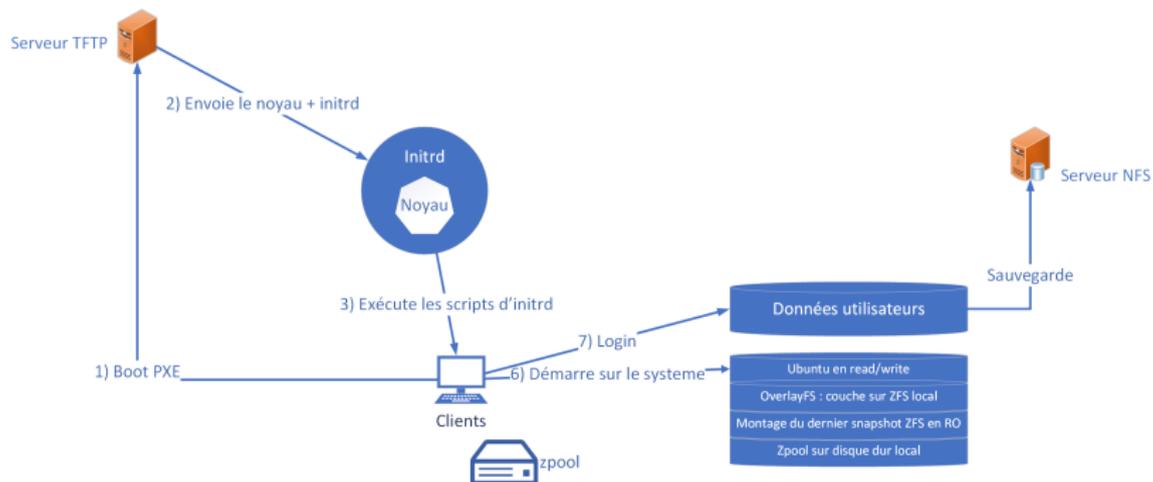
# Diskfull à l'IMB

# Besoins

- ▶ Gestion centralisée à la manière de *FaDDeF*
- ▶ Minimiser les hacks/tricks pour adapter la distribution au fonctionnement
- ▶ Sortir le Home utilisateur du NFS (problèmes avec Thunderbird et Firefox)
- ▶ Éloignement de la salle serveur (datacenter) → NFS ne supportera pas

⇒ Utilisation du disque local. On profite ainsi des SSD/NVME. Choix de ZFS pour le système de fichier.

# Fonctionnement



# Fonctionnement

1. Allumage de l'ordinateur
2. Configuration IP par DHCP, démarrage réseau avec PXE
3. Récupération du noyau et d'une image initrd personnalisée par TFTP/HTTP
  - ▶ premier démarrage : formatage du disque puis synchronisation du système sur le disque via ZFS send/receive
  - ▶ démarrages suivants : montage du dernier snapshot ZFS en ReadOnly, mise en place de l'OverlayFS
4. Lancement des services systemd
5. Ouverture de session par l'utilisateur

Les applications s'exécutent sur le CPU et la mémoire de l'ordinateur à partir des programmes et fichiers disponibles sur le disque local. Un service *systemd* se déclenche à intervalle régulier pour synchroniser sur le disque local les nouveaux snapshots disponibles sur le serveur central.



# Construction du serveur ZFS qui héberge la distribution

- ▶ Création d'un dataset ZFS
- ▶ Debootstrap d'Ubuntu dans le dataset ZFS
- ▶ Construction d'une image initrd spécifique à l'aide des outils *initramfs-tools* en lui ajoutant trois scripts développés à l'IMB :
  - ▶ *01\_imb\_check* → vérification du disque et de la présence des différents ZFS (ubuntu, overlaysfs, home)
  - ▶ *02\_imb\_sync* → synchronisation ZFS de la distribution Ubuntu du serveur vers le disque local
  - ▶ *03\_config* → Monte le dernier snapshot ZFS en read-only, met en place l'OverlayFS, démarre le système sur cet espace.
- ▶ Mise en place du service Webhook pour répondre aux demandes des clients : obtenir le nom du dernier snapshot, demander une synchro ZFS au serveur.



# Installation d'une nouvelle machine

1. Déballer et brancher la machine
2. Configuration DHCP/DNS
3. configuration du BIOS/UEFI pour démarrer en PXE
4. choix du mode "Initialisation Diskfull"
5. La machine redémarre en mode Diskfull classique (choix par défaut)



# Remplacement d'une machine

1. Reprendre les étapes d'installation d'une machine
2. Restauration des données utilisateurs à l'aide de Restic

# Emplacement des données utilisateurs

- ▶ Le home est stocké dans un volume ZFS sur le disque local
- ▶ ZFS apporte la gestion des quotas, les snapshots (avec ZnapZend) et le chiffrement
- ▶ Sauvegarde automatique du home sur un espace NFS par l'outil *restic*
- ▶ Mise à disposition d'un espace NFS pour partager des données avec un autre ordinateur sur le réseau, ou avec des serveurs
- ▶ Utilisation de solution cloud (plmbox) et de Git pour le partage de données entre utilisateurs



# Prise en main

- ▶ Où trouver la documentation ?
  - ▶ [https://plmlab.math.cnrs.fr/sylvain.allemand/diskfull\\_zfs](https://plmlab.math.cnrs.fr/sylvain.allemand/diskfull_zfs)
- ▶ Technologies / pré-requis à maîtriser :
  - ▶ ZFS
  - ▶ SystemD
  - ▶ Linux

# Postes fixes au LAREMA

# Besoins

- ▶ Postes de travail banalisés que l'on peut déplacer d'un bureau à l'autre sans modification de configuration ni recopie de fichiers utilisateurs
- ▶ Postes non affectés : les utilisateurs retrouvent leurs configurations et données peu importe le poste qu'ils utilisent. Les données ne sont pas stockées localement.
- ▶ Système strictement identique sur tous les postes, sans divergence possible au fil du temps.

# Fonctionnement

2 modes de fonctionnement :

1. diskless : fonctionne avec un / distant, accédé via NFS en read-only, avec des parties (fichiers ou répertoires) en read-write gérés via `readonly-root` (voir `stateless linux`)
2. local / : fonctionne avec un / local (au choix en read-only comme précédemment ou en read-write)

# Fonctionnement

- ▶ client lourd : les applications, distantes en mode 1 ou locales en mode 2, s'exécutent toujours sur le CPU et la mémoire du poste
- ▶ un poste peut synchroniser (en tâche de fond) le / distant sur une partition locale (s'il est en mode 2, il modifie donc la partition système en cours d'utilisation)
- ▶ le fonctionnement d'un poste dépend de variables passées au kernel et récupérées via PXE, comme par exemple :
  - ▶ le mode de fonctionnement du poste (1 ou 2)
  - ▶ l'URL du / distant (pour booter en mode 1, ou synchroniser en mode 1 et 2)
  - ▶ le / en read-only ou read-write
  - ▶ activation ou non de la synchronisation

# Fonctionnement

1. Allumage de l'ordinateur
2. Configuration IP par DHCP
3. Récupération du kernel et de ses paramètres via PXE
4. Montage du / (NFS si mode 1, partition locale si mode 2)
5. Montage NFS des données de l'utilisateur

# Installation d'une nouvelle machine

1. Déballer et brancher la machine, configurer le BIOS pour booter en réseau
2. La déclarer dans le DNS et DHCP, la configurer en mode 1
3. Allumer la machine

Le poste est alors opérationnel en mode diskless.

- ▶ Pour qu'il soit apte au mode 2 (ce qui n'est pas obligatoire), il suffit de partitionner le disque local de façon adéquate (en lançant un simple shell)
- ▶ Il sera réellement apte au mode 2 après une première synchronisation (il suffit d'attendre)



# Remplacement d'une machine

1. Retirer l'ancienne machine
2. Installer la nouvelle, comme expliqué précédemment

# Environnement

- ▶ nécessité de disposer
  1. d'1 ou 2 serveurs NFS (système+homes)
  2. d'un serveur DNS+DHCP+tftp
- ▶ l'image système de référence est fabriquée sur une machine indépendante autonome via des playbooks ansible



# Prise en main

- ▶ Où trouver la documentation
  - ▶ la documentation est à faire
- ▶ Technologies / pré-requis à maîtriser ?
  - ▶ lowtech (dns-dhcp-tftp-nfs) + ansible

# Kit Ubuntu

# Besoins

- ▶ Pour quoi faire ?
  - ▶ Automatiser l'installation du système Ubuntu
  - ▶ Industrialiser la configuration des machines (tendre vers du *zéro touch*)
  - ▶ Spécialisation de machines (pilotes, logiciels)
- ▶ Sur quels type de matériel ?
  - ▶ Des postes lourds<sup>3</sup>
  - ▶ Principalement des ordinateurs fixes mais aussi portables
  - ▶ Pour des salles de TP et laboratoire de recherche
  - ▶ Avec double amorçage (Windows + GNU/Linux)



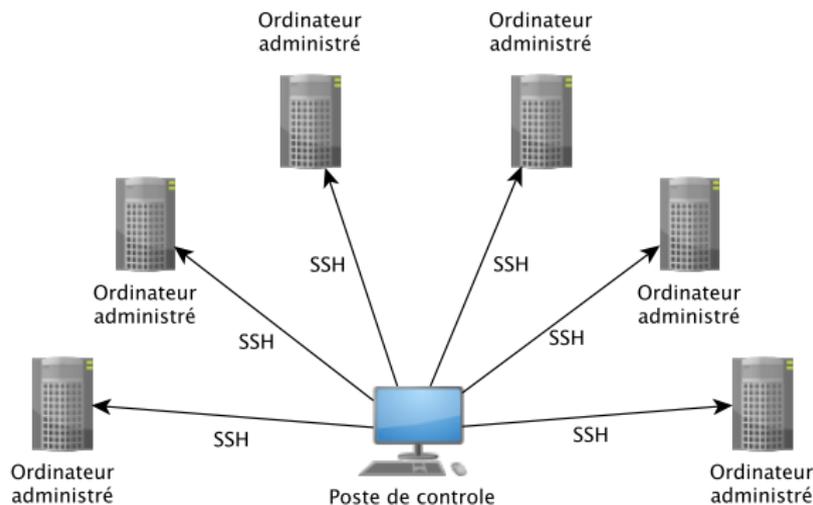
Un poste lourd est une machine sur laquelle s'exécute la totalité des applications sans utilisation de ressources externes

# Fonctionnement

1. Obtention automatique de la configuration réseau par DHCP
2. Chargement du noyau et de l'image initrd avec l'installeur Ubuntu Ubiquity (TFTP)
3. Montage d'une copie de l'image iso Ubuntu Desktop par NFS
4. Exécution de l'installeur Ubiquity Ubuntu
  - ▶ Réponses aux questions obtenues depuis le fichier de réponse (preseed)
  - ▶ Seul le partitionnement doit être effectué par l'opérateur
5. Installation d'un système Ubuntu minimal avec interface graphique sur le disque de la machine
6. Redémarrage automatique en fin d'installation
7. Post-installation par Ansible



# Environnement



Le poste de contrôle porte les services de l'installation automatisée (DHCP, TFTP, NFS). Il est également le chef d'orchestre qui agit sur les machines du parc via Ansible.



# Ansible

L'outil ansible est utilisé pour la post-installation :

- ▶ renommer l'ordinateur, sécuriser GRUB
- ▶ configuration DNS et proxy-cache
- ▶ synchronisation horaire NTP
- ▶ routage des emails système, journalisation
- ▶ pilotes périphériques spécifiques (claviers, cartes graphiques GPU)
- ▶ définition du serveur d'impression (CUPS)
- ▶ mises à jour automatique
- ▶ authentification et identification des utilisateurs
  - ▶ jonction d'un domaine ActiveDirectory
  - ▶ annuaire LDAP et royaume Kerberos (type MIT)
  - ▶ comptes locaux
- ▶ installation de paquets supplémentaires et de logiciels hors distribution



# Ansible

L'outil ansible est utilisé pour la post-installation :

- ▶ renommer l'ordinateur, sécuriser GRUB
- ▶ configuration DNS et proxy-cache
- ▶ synchronisation horaire NTP
- ▶ routage des emails système, journalisation
- ▶ pilotes périphériques spécifiques (claviers, cartes graphiques GPU)
- ▶ définition du serveur d'impression (CUPS)
- ▶ mises à jour automatique
- ▶ authentification et identification des utilisateurs
  - ▶ jonction d'un domaine ActiveDirectory
  - ▶ annuaire LDAP et royaume Kerberos (type MIT)
  - ▶ comptes locaux
- ▶ installation de paquets supplémentaires et de logiciels hors distribution



Installation et configuration complète du poste de contrôle par Ansible.

# Installation d'une nouvelle machine

1. Déballer et brancher la machine
2. Lui attribuer un nom et une adresse IP
3. Renseigner le couple adresses MAC + IP dans le serveur DHCP
4. Compléter le serveur DNS
5. (Installer le système de Microsoft si nécessaire avec WDS)
6. Allumer la machine et choisir un amorçage réseau
7. Installation automatisée du système Ubuntu
8. Post-installation avec Ansible



# Remplacement d'une machine

1. Installation du système comme pour une nouvelle machine
2. Reprise des données utilisateurs :
  - ▶ Si /home sur NFS → rien à faire
  - ▶ Si /home sur disque local et qu'il est toujours fonctionnel → copie de la partition via *rsync*
  - ▶ Si /home sur disque local et qu'il est en panne → reprise des données depuis la sauvegarde



# Emplacement des données utilisateurs

- ▶ Montage du répertoire d'accueil par NFS
- ▶ Partition du disque local dédiée au /home (création automatique du répertoire d'accueil à la première connexion)



# Prise en main

- ▶ Un kit de déploiement prêt à l'emploi disponible sur <https://src.koda.cnrs.fr/benoit.metrot.2/ubuntu-toolkit/>
- ▶ Une formation de 3 jours dispensée à l'Université de Poitiers en 2022 et 2023
- ▶ Un article aux JRES de 2024 à Rennes  
<https://2024.jres.org/programme#modal-46>

# Synthèse

# Clavier anglais

Est-il possible d'avoir un clavier anglais sur certaines machines ?

- ▶  FaDDeF  
Tricks de détection du clavier par l'identifiant USB
- ▶  Diskfull à l'IMB  
Tricks de détection du clavier par l'identifiant USB
- ▶  Postes fixes au LAREMA
- ▶  Kit Ubuntu  
Spécialisation avec les variables de groupes Ansible

# Postes autonomes

Les postes peuvent-ils fonctionner en l'absence de serveurs du réseau (NFS par exemple) ?

- ▶ ❌ FaDDeF, Diskless (DHCP + PXE + NFS)
- ▶ ❌ Diskfull à l'IMB (DHCP + PXE)
- ▶ ❌ Postes fixes au LAREMA
- ▶ ✅ Kit Ubuntu

# Types de postes : machine de laboratoire

L'outil est-il adapté pour déployer le système sur des machine de laboratoire avec des logiciels différents pour chaque personne ?

- ▶ ❌ FaDDeF, Diskless
- ▶ ✅ Diskfull à l'IMB
- ▶ ❌ Postes fixes au LAREMA
- ▶ ✅ Kit Ubuntu

# Types de postes : salles de TP

L'outil est-il adapté pour déployer le système sur des machines homogènes dans des salles de travaux pratiques ?

- ▶ ✓ FaDDeF, Diskless
- ▶ ✓ Diskfull à l'IMB
- ▶ ✓ Postes fixes au LAREMA
- ▶ ✓ Kit Ubuntu

# Type de postes : portables

Les ordinateurs portables sont-ils supportés ?

- ▶ ❌ FaDDeF
- ▶ ❌ Diskfull à l'IMB
- ▶ ❌ Postes fixes au LAREMA
- ▶ ✅ Kit Ubuntu

# Propagation des modifications

Les modifications sont-elles appliquées au fil de l'eau (sans intervention) ?

- ▶ ✓ FaDDeF
- ▶ ✓ Diskfull à l'IMB (redémarrage nécessaire)
- ▶ ✓ Postes fixes au LAREMA
- ▶ ✗ Kit Ubuntu

Seulement quand l'administrateur décide de l'application d'un playbook

# Délai d'attente pour remplacer le poste d'un utilisateur

Quel est le temps nécessaire pour remplacer le poste d'un utilisateur tombé en panne ?

- ▶ FaDDeF : instantané !
- ▶ Diskfull à l'IMB : temps de synchro du système (5min) + temps de restauration Restic
- ▶ Postes fixes au LAREMA :
- ▶ Kit Ubuntu : temps de réinstallation d'une machine (30min environ, dépend du nombre de logiciels) + temps de recopie des données si /home sur disque local