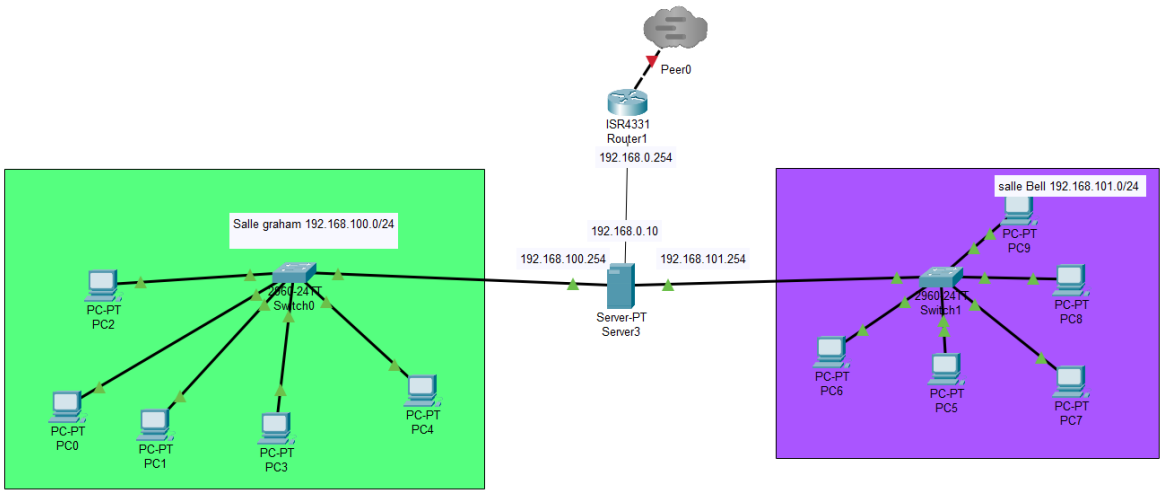
Projet : Automatisez la gestion d'un Parc Informatique

(Documentation)

Sommaire

Présentation générale du réseau



1. Salles (Subnets) :

- Salle Graham : Avec l'adresse réseau 192.168.100.0/24, contenant 5 ordinateurs (PC0 à PC4) connectés à un commutateur (Switch0).
- Salle Bell : Avec l'adresse réseau 192.168.101.0/24, contenant 4 ordinateurs (PC5 à PC9) connectés à un autre commutateur (Switch1).

2. Routeur (Router1) :

L'appareil nommé ISR4331 porte l'adresse IP 192.168.0.254 et est le point de passage entre les deux sous-réseaux locaux et une connexion externe représentée par le nuage avec le label Peer0.

3. Serveur (Server3) :

- Se trouve entre les deux sous-réseaux, avec deux interfaces réseau :
    - Une dans la Salle Graham avec l'adresse 192.168.100.254.
    - Une dans la Salle Bell avec l'adresse 192.168.101.254.

Ce serveur peut servir à différentes fins, comme un serveur de fichiers, un serveur d'applications, ou un serveur TFTP comme mentionné précédemment.

4. Connectivité :

- Chaque PC dans les sous-réseaux est connecté à son commutateur local, qui est ensuite connecté au routeur. Ce schéma permet à chaque PC d'accéder aux ressources de l'autre subnet via le routeur et potentiellement à Internet ou à d'autres réseaux via le nuage Peer0.
- Le routeur dispose également d'une adresse dans un troisième réseau 192.168.0.10, qui pourrait être utilisé pour la gestion du routeur ou pour connecter à un autre segment réseau non représenté ici.

5. IP Addressing :

Les adresses IP des sous-réseaux sont dans la plage privée, ce qui est typique pour les réseaux internes. Le masque de sous-réseau /24 indique que chaque salle peut accueillir jusqu'à 254 appareils (256 adresses au total moins l'adresse réseau et l'adresse de broadcast).

Les Scripts :

Script pour l'installation complète des postes (langue, fuseau horaire, mot de passe, comptes, dépôts, interface graphique…) via le réseau :

Ce script est un fichier de préconfiguration (preseed) pour l'installation automatisée de Debian, une distribution Linux populaire. Le but de ce fichier est de fournir à l'installateur Debian (debian-installer) toutes les réponses aux questions qui sont normalement posées pendant une installation interactive. Cela permet une installation sans surveillance, ce qui est particulièrement utile pour déployer plusieurs machines de manière homogène et efficace. Voici un résumé de ce que le script configure, organisé par sections :

Localisation

Langue, pays, et locale : Le français (France) est sélectionné avec l'encodage UTF-8 (fr_FR.UTF-8).

Clavier : Le layout du clavier est configuré pour le français.

Configuration réseau

Activation du réseau : Le réseau est activé et configuré pour utiliser DHCP pour obtenir automatiquement une adresse IP, un masque de sous-réseau, une passerelle et des serveurs DNS. Il y a des options pour configurer un réseau statique, mais elles sont commentées et donc non utilisées.

Paramètres du miroir Debian

Miroir HTTP : Les paquets seront téléchargés depuis le miroir http.us.debian.org pour l'installation.

Configuration des comptes

Mot de passe root : Le mot de passe du superutilisateur (root) est défini comme toor.

Création d'un utilisateur : Un compte d'utilisateur standard est créé avec le nom d'utilisateur debian et le mot de passe insecure. Cet utilisateur est ajouté aux groupes audio, cdrom, et video.

Configuration de l'horloge et du fuseau horaire

Horloge UTC : L'horloge du matériel (hardware clock) est configurée pour utiliser l'heure UTC.

Fuseau horaire : Le fuseau horaire est défini sur Europe/Paris.

NTP : L'heure sera synchronisée avec un serveur NTP.

Partitionnement

Méthode de partitionnement : Utilisation de LVM pour le partitionnement avec une taille maximale pour le groupe de volumes.

Schéma de partitionnement : Une recette de partitionnement atomic est sélectionnée, ce qui crée une seule grande partition pour tout, à l'exception d'une petite partition de swap.

Installation du système de base

Sélection de paquets : Installation des tâches standard, web-server, et desktop. De plus, openssh-server, build-essential, et TFTP sont spécifiquement inclus.

Installation du chargeur de démarrage

GRUB : Le chargeur de démarrage GRUB est installé et configuré pour démarrer Debian par défaut, même si d'autres systèmes d'exploitation sont détectés.

Fin de l'installation

Redémarrage automatique : La machine redémarrera automatiquement une fois l'installation terminée.

Exécution de commandes personnalisées

Commandes tardives : Installation du paquet sudo juste avant la fin de l'installation via apt-install.

Ce script de préconfiguration est conçu pour une installation Debian minimale mais fonctionnelle, avec un accent particulier sur la facilité d'utilisation et une certaine flexibilité dans la configuration réseau. La possibilité d'ajouter des utilisateurs, de configurer l'heure, et d'installer des paquets supplémentaires montre une volonté d'obtenir un système prêt à l'emploi dès le premier démarrage. Pour reprendre ou modifier ce travail, un administrateur devrait comprendre les bases de la configuration réseau, du système de fichiers Linux, de la gestion des paquets Debian, ainsi que des principes de sécurité de base tels que la configuration des mots de passe et des groupes d'utilisateurs. Pour finir des commentaires sont aussi présents dans le script afin de vous guider au mieux dans la suite de votre travail.

Script pour la configuration des machines (nom de machine unique, partage NFS avec raccourci sur le bureau, serveur ssh, serveur vnc, proxy apt, connexion par clé de sécurité...) :

1. hostname.sh

Objectif : Configurer le nom d'hôte de la machine basé sur son adresse IP.

Fonctionnement détaillé :

Le script commence par obtenir une adresse IP via DHCP, assurant que la machine dispose d'une adresse IP valide.

Il extrait ensuite les deux derniers chiffres de cette adresse IP pour les utiliser dans le nouveau nom d'hôte.

Le nom d'hôte est défini en utilisant hostnamectl set-hostname, un outil standard pour changer le nom d'hôte sur les systèmes Linux modernes.

Pour garantir la cohérence du système, le script met à jour le fichier /etc/hosts, en supprimant l'ancienne entrée 127.0.1.1 et en ajoutant une nouvelle entrée pour le nouveau nom d'hôte.

Implications :

Utiliser l'adresse IP pour générer le nom d'hôte assure l'unicité et la traçabilité des noms d'hôte dans un réseau.

La mise à jour de /etc/hosts est cruciale pour que le système résolve correctement son propre nom d'hôte en adresse IP, évitant ainsi d'éventuels problèmes de réseau ou d'application.

2. menu.sh

Objectif : Offrir un menu interactif pour exécuter divers scripts d'administration.

Fonctionnement détaillé :

- Le script affiche une liste d'options, chacune correspondant à un autre script ou à l'option de quitter.
- L'utilisateur sélectionne une option par son numéro.
- Selon le choix, le script correspondant est rendu exécutable avec chmod +x et exécuté.

Implications :

- Centralise l'accès à divers outils d'administration, simplifiant le processus pour les administrateurs systèmes.
- Assure que les scripts sont exécutables avant leur lancement, réduisant les erreurs potentielles dues à des permissions de fichier incorrectes.

3. vnc.sh

Objectif : Installer et configurer un serveur VNC avec Xfce comme environnement de bureau.

Fonctionnement détaillé :

Le script met à jour la liste des paquets et installe les paquets nécessaires pour VNC et Xfce.

Il demande ensuite le nom d'utilisateur pour lequel configurer le serveur VNC.

Si l'utilisateur n'existe pas, il est créé et configuré.

Enfin, il lance vncserver pour initialiser la configuration VNC de l'utilisateur.

Implications :

- Permet un accès à distance au bureau Linux, facilitant la gestion à distance.
- L'installation de Xfce offre une expérience de bureau légère et réactive pour les connexions VNC.

4. nfs.sh

Objectif : Monter un répertoire partagé NFS et créer un raccourci sur le bureau.

Fonctionnement détaillé :

- Détermine l'adresse du serveur NFS et le chemin du répertoire à monter.
- Crée le point de montage local si nécessaire et monte le répertoire partagé.
- Si possible, crée un raccourci sur le bureau de l'utilisateur pour accéder facilement au répertoire NFS.

Implications :

- Facilite l'accès aux fichiers partagés sur le réseau, rendant le travail collaboratif plus efficace.
- La création d'un raccourci sur le bureau améliore l'expérience utilisateur, rendant l'accès au partage NFS transparent.

5. key.sh

Objectif : Automatiser la création d'un nouvel utilisateur sur des postes clients et configurer un accès SSH sans mot de passe à l'aide de clés SSH.

Fonctionnement :

- Génération de clés SSH : Vérifie si une paire de clés SSH existe déjà pour l’administrateur ; sinon, elle est générée.
- Création de l'utilisateur : Se connecte à chaque poste client spécifié, crée un nouvel utilisateur et le configure pour un accès sans mot de passe en ajoutant la clé publique de l'administrateur au fichier authorized_keys de l'utilisateur.

Sécurité : L'accès SSH sans mot de passe est pratique mais nécessite que les clés SSH soient stockées et gérées de manière sécurisée. L'utilisation de phrases secrètes avec les clés SSH et la restriction des permissions sur le fichier authorized_keys sont des pratiques recommandées.

Script pour la configuration de leur administration à distance via le serveur :

Ce script est un outil d'administration à distance pour des postes Linux, utilisant principalement SSH pour exécuter des commandes sur des machines distantes. Voici une documentation détaillée de ses fonctions et de leur utilisation :

Fonctions disponibles

- Redémarrage d'un poste (reboot_poste) : Redémarre une machine distante à travers SSH.
- Arrêt d'un poste (shutdown_poste) : Éteint une machine distante à travers SSH.
- Changement du nom de la machine (changer_nom_machine) : Modifie le nom d'hôte de la machine distante et son nom dans le fichier /etc/hosts, puis redémarre la machine pour appliquer le changement.
- Création d'un utilisateur (creer_utilisateur) : Ajoute un nouvel utilisateur au système distant via SSH.
- Réinitialisation d'un poste (reinitialiser_poste) : Purge tous les paquets installés sur une machine distante, réinitialise le fichier /etc/passwd (danger : cette opération supprime tous les utilisateurs !), réinstalle sudo, puis redémarre la machine. Cette fonction est utile pour remettre une machine à un état "propre" en fin de formation ou avant une réaffectation.
- Vérification de la connectivité Internet (check_internet) : Teste si la machine distante peut accéder à Internet en pingant l'adresse 8.8.8.8.
- Installation d'un paquet (installer_paquet) : Installe un paquet sur la machine distante si celle-ci a accès à Internet. En cas d'échec de la connexion Internet, le script propose de tenter de récupérer une adresse IP via DHCP et de réessayer.
- Connexion SSH (ssh_connect) : Permet de se connecter via SSH à une machine distante.
- Exécution d'une commande sur un ensemble de postes (executer_sur_ensemble) : (Non implémentée) Serait utilisée pour exécuter une

commande sur plusieurs postes à la fois en lisant leurs adresses IP depuis un fichier.

Menu principal

Le script propose un menu interactif offrant les options suivantes :

1. Redémarrer un poste
2. Éteindre un poste
3. Changer le nom de la machine
4. Créer un utilisateur
5. Réinitialiser un poste en fin de formation
6. Installer un paquet
7. Se connecter en SSH
8. Exécuter une commande sur l'ensemble des postes simultanément
9. Quitter

L'utilisateur doit choisir une option en entrant le numéro correspondant. Pour chaque action, le script demandera des informations supplémentaires, telles que l'adresse IP de la machine distante et, selon l'option choisie, le nouveau nom de l'hôte, le nom d'utilisateur à créer, ou le nom du paquet à installer.

Utilisation

Pour utiliser ce script :

1. Assurez-vous que l'utilisateur administrateur existe sur toutes les machines distantes et a les privilèges nécessaires pour exécuter des commandes sudo sans mot de passe.
2. Le script doit être exécuté sur une machine Linux avec SSH installé et configuré.
3. Les machines cibles doivent être accessibles sur le réseau, avec SSH activé et configuré pour accepter les connexions de l'utilisateur administrateur.

Ce script facilite la gestion de plusieurs postes Linux à distance, en automatisant des tâches courantes d'administration système. Il est cependant crucial de l'utiliser avec prudence, surtout avec des fonctions comme la réinitialisation d'un poste, qui supprime toutes les données utilisateur et les configurations. De même pour ce script-ci des

commentaires sont aussi présents afin de vous guider au mieux dans la suite de votre travail.

VNC Server :

Sur un serveur VNC (Virtual Network Computing), l'objectif est de permettre l'accès à distance à l'interface graphique d'un système. Les services et paquets suivants jouent des rôles importants dans la mise en place et l'enrichissement de cette expérience utilisateur. Voici une liste de tous les services installés sur ce serveur :

tightvncserver

TightVNC Server est une implémentation de VNC qui permet de contrôler à distance des machines via un réseau. Il s'agit d'un logiciel qui exécute un serveur VNC, lequel écoute les connexions entrantes de clients VNC. Une fois connecté, l'utilisateur peut voir l'écran du serveur VNC et interagir avec lui, comme s'il était assis devant. TightVNC est particulièrement apprécié pour ses performances optimisées pour les connexions lentes, grâce à une compression efficace des données d'image. Il permet aussi une configuration avancée de la sécurité, comme le cryptage des sessions et l'authentification par mot de passe.

xfce4

Xfce4 est un environnement de bureau léger pour les systèmes UNIX-like, y compris Linux. Sa légèreté en fait un choix idéal pour les serveurs VNC, surtout sur les systèmes avec des ressources limitées. Xfce fournit une interface utilisateur graphique (GUI) simple, intuitive, et modulable, incluant un gestionnaire de fenêtres, un panneau pour l'accès aux applications et configurations, un bureau avec des icônes, et un gestionnaire de fichiers. Il vise à être rapide et consommer peu de ressources système tout en restant visuellement attrayant et facile à utiliser.

xfce4-goodies

Xfce4-goodies est un ensemble de logiciels supplémentaires et d'extensions pour l'environnement de bureau Xfce. Ce paquet comprend une variété de plugins, de

widgets (comme des moniteurs système, des gestionnaires de tâches, des calendriers), et des thèmes qui peuvent être ajoutés au panneau Xfce ou au bureau. Les "goodies" enrichissent l'expérience utilisateur en offrant des fonctionnalités supplémentaires et des options de personnalisation sans compromettre la performance de l'environnement de bureau.

xfce4-weather-plugin

Xfce4-weather-plugin est un plugin spécifique pour le panneau Xfce qui affiche des informations météorologiques dans le panneau de l'environnement de bureau Xfce. Il peut montrer les conditions actuelles, les prévisions, et divers détails météorologiques comme la température, la vitesse du vent, l'humidité, etc., en se basant sur les données de différentes stations météorologiques à travers le monde. C'est un exemple de la manière dont les "goodies" peuvent améliorer l'expérience utilisateur sur un bureau Xfce en fournissant des informations utiles directement sur le bureau.

Serveur NFS (Debian 11) :

NFS est un protocole qui permet à des utilisateurs sur un réseau informatique d'accéder à des fichiers situés sur un autre ordinateur de la même manière que s'ils étaient stockés localement.

nfs-kernel-server

Le service nfs-kernel-server sur un serveur NFS (Network File System) sous Debian 11 joue un rôle central dans la mise en œuvre du partage de fichiers en réseau. NFS est un protocole qui permet à des utilisateurs sur un réseau informatique d'accéder à des fichiers situés sur un autre ordinateur de la même manière que s'ils étaient stockés localement. Voici les principaux aspects et fonctionnalités du nfs-kernel-server :

Rôle du nfs-kernel-server

- Partage de répertoires : nfs-kernel-server gère le partage des répertoires sur le serveur. Il définit quels répertoires sont partagés, avec qui et avec quelles permissions. Cette configuration est généralement définie dans le fichier /etc/exports.

- Gestion des demandes client : Il traite les demandes d'accès aux fichiers venant des clients NFS. Cela comprend la lecture, l'écriture, la création et la suppression de fichiers ou de dossiers dans les répertoires partagés.
- Sécurité : nfs-kernel-server permet de configurer des mesures de sécurité pour le partage de fichiers. Cela inclut la limitation des accès en fonction des adresses IP ou des réseaux, ainsi que la configuration des permissions au niveau des fichiers et des répertoires partagés.
- Optimisation des performances : Le serveur NFS utilise le noyau Linux pour gérer le partage de fichiers, ce qui peut offrir de meilleures performances comparées à certaines solutions basées sur l'espace utilisateur. Il prend en charge des fonctionnalités avancées comme NFSv4, le partage de fichiers à travers des réseaux dans des environnements hétérogènes (Linux, UNIX, Windows).

Configuration et gestion

- Configuration des partages : L'administrateur système configure les partages NFS en éditant le fichier /etc/exports pour déterminer quels répertoires sont partagés, avec quels hôtes ou réseaux, et avec quelles options (comme la lecture seule, la lecture-écriture, etc.).
- Contrôle d'accès : Il est possible de définir des contrôles d'accès fins pour chaque partage, limitant l'accès à certains utilisateurs ou groupes et définissant les types d'opérations autorisées.
- Démarrage et arrêt du service : nfs-kernel-server peut être démarré, arrêté, et redémarré à l'aide des commandes système appropriées (par exemple, systemctl start nfs-kernel-server sous Debian 11), permettant une gestion flexible des partages NFS.

Serveur TFTP :

Un serveur TFTP (Trivial File Transfer Protocol) et SSHFS (SSH Filesystem) fournissent des moyens de transférer et de partager des fichiers sur un réseau, chacun ayant des utilisations et des caractéristiques spécifiques. Voici une explication détaillée de ces services :

Serveur TFTP

TFTP est une version simplifiée du protocole FTP (File Transfer Protocol) conçue pour transférer des fichiers de manière simple sans authentification ni chiffrement. TFTP est largement utilisé dans des scénarios où la simplicité et l'efficacité sont prioritaires, comme le déploiement de logiciels sur des appareils réseau (routeurs, switchs) ou des systèmes embarqués, ainsi que le démarrage de machines sans disque (PXE boot).

Fonctionnalités et utilisation de TFTP :

- Transfert de fichiers sans authentification : TFTP permet le transfert de fichiers sans nécessiter de nom d'utilisateur ou de mot de passe, ce qui facilite les opérations dans des environnements contrôlés.
- Simplicité et efficacité : Avec un fonctionnement sur UDP, TFTP est conçu pour être léger et rapide, bien adapté pour transférer de petits fichiers de configuration ou des images de firmware.
- Utilisation en réseau local : En raison de l'absence de chiffrement et d'authentification, TFTP est généralement utilisé dans des réseaux locaux sécurisés ou pour des applications spécifiques où la sécurité n'est pas une préoccupation majeure.

SSHFS

SSHFS permet de monter à distance des systèmes de fichiers à travers une connexion SSH (Secure Shell), fournissant un accès sécurisé aux fichiers sur un serveur distant. SSHFS utilise le protocole SFTP (SSH File Transfer Protocol), une extension de SSH, pour réaliser les opérations de fichier.

Fonctionnalités et utilisation de SSHFS :

- Accès sécurisé aux fichiers : SSHFS chiffre toutes les communications entre le client et le serveur, protégeant les données contre les écoutes indiscrètes.
- Facilité d'utilisation : Avec SSHFS, les utilisateurs peuvent monter un répertoire distant dans leur système de fichiers local comme s'il était un disque dur ou un partage réseau local, facilitant l'accès et la manipulation des fichiers distants.

Serveur SSH :

Le service openssh-server sur un serveur SSH (Secure Shell) sous Debian 11 joue un rôle crucial dans la sécurisation des communications réseau. OpenSSH est un ensemble d'outils de connectivité réseau utilisés pour accéder à des sessions à distance sécurisées entre machines sur un réseau. L'installation du paquet openssh-server permet à une machine Debian d'accepter les connexions entrantes SSH, offrant ainsi une méthode sécurisée pour exécuter des commandes à distance, transférer des fichiers, et gérer le système.

Principales fonctionnalités d'OpenSSH Server

- Accès à distance sécurisé : OpenSSH permet aux utilisateurs autorisés d'accéder à une machine à distance, exécuter des commandes, et manipuler des fichiers. Toutes les données (y compris les mots de passe) sont chiffrées, ce qui prévient l'écoute clandestine et le vol d'informations.

- Authentification flexible : openssh-server supporte plusieurs méthodes d'authentification, y compris l'authentification par mot de passe, la clé publique/privée, et l'authentification par Kerberos. Cela permet aux administrateurs de choisir le niveau de sécurité adapté à leur environnement.

- Tunneling SSH (Port Forwarding) : SSH peut encapsuler d'autres protocoles (comme HTTP, FTP, etc.) à travers une connexion chiffrée. Ce tunneling permet de sécuriser la communication entre le client et le serveur pour des applications non sécurisées et de contourner des pares-feux ou des filtres.

- X11 Forwarding : Permet d'exécuter des applications graphiques sur le serveur et de les afficher sur le poste client, à travers une connexion SSH sécurisée. Cela est utile pour des tâches d'administration système nécessitant une interface graphique.

- SFTP/SCP : OpenSSH inclut des protocoles de transfert de fichiers (SFTP et SCP) pour un transfert sécurisé des données. Ces protocoles utilisent le même niveau de chiffrement que la session SSH, assurant la sécurité des fichiers transférés.

Avantages de l'utilisation d'OpenSSH Server

- Sécurité : OpenSSH est une solution de sécurité éprouvée pour l'accès à distance, fournissant un chiffrement fort pour protéger contre l'écoute clandestine et les attaques de l'homme du milieu.

- Interopérabilité : OpenSSH est compatible avec une grande variété de systèmes d'exploitation, y compris toutes les distributions Linux, macOS, et Windows (avec le client SSH intégré ou des tiers comme PuTTY).

- Gestion centralisée : Permet aux administrateurs de gérer de manière centralisée les machines, les utilisateurs, et les clés, facilitant la maintenance et l'administration de systèmes distants.

- Réduction des coûts : Comme OpenSSH est un logiciel libre et open source, il offre une solution puissante et flexible sans les coûts associés à des logiciels commerciaux.