

# Projet Linux – Automatisation de la gestion des utilisateurs et groupes

Simplification et efficacité dans la  
gestion des comptes Linux







# Présentation du projet Linux Ubuntu

## **Objectif du projet**

Automatiser la gestion des utilisateurs et groupes via des scripts Bash pour simplifier l'administration système.

## **Fonctionnalités des scripts**

Deux scripts distincts gèrent la création et la suppression des utilisateurs et groupes avec commandes système.

## **Apprentissage et pratiques**

Projet introduit les concepts d'administration Linux et les bonnes pratiques de scripting Bash pour les étudiants.

# OBJECTIFS DU PROJET

- Automatiser la **gestion des utilisateurs et des groupes** via des scripts Bash.
- Proposer un **menu interactif** pour faciliter les actions de création et de

suppression.

Le script demande :

« **Voulez-vous ajouter cet utilisateur à un groupe existant ? (o/n) »** »

Contrôler l'existence des utilisateurs et groupes avant chaque opération.

Deux cas :

- **o** → on demande le nom du groupe.
- S'il existe → ajout.
- Sinon → erreur.
- **n** → on passe directement au menu.

Dans tous les cas :

 **retour au menu principal** (fin locale).

# Docker pourquoi et comment

## Objectif du projet

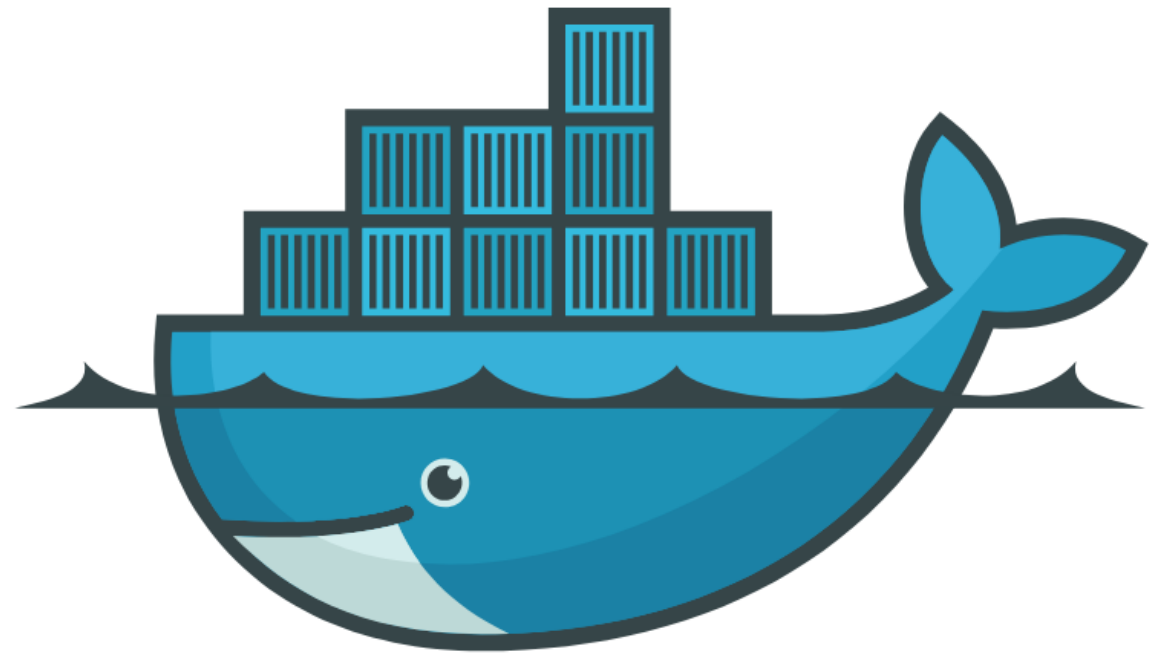
Automatiser la gestion des utilisateurs et groupes via des scripts Bash pour simplifier l'administration système.

## Fonctionnalités des scripts

Deux scripts distincts gèrent la création et la suppression des utilisateurs et groupes avec commandes système.

## Apprentissage et pratiques

Projet introduit les concepts d'administration Linux et les bonnes pratiques de scripting Bash pour les étudiants.



# docker

# SCRIPT DE CRÉATION

## Fonctionnement du script de création

## Validation des noms

Le script vérifie que les noms d'utilisateurs et de groupes respectent un format alphanumérique avec tirets et underscores.

## Menu interactif

Une boucle infinie propose un menu interactif permettant de choisir la création d'un utilisateur, d'un groupe ou de quitter le programme.

## Création de groupes

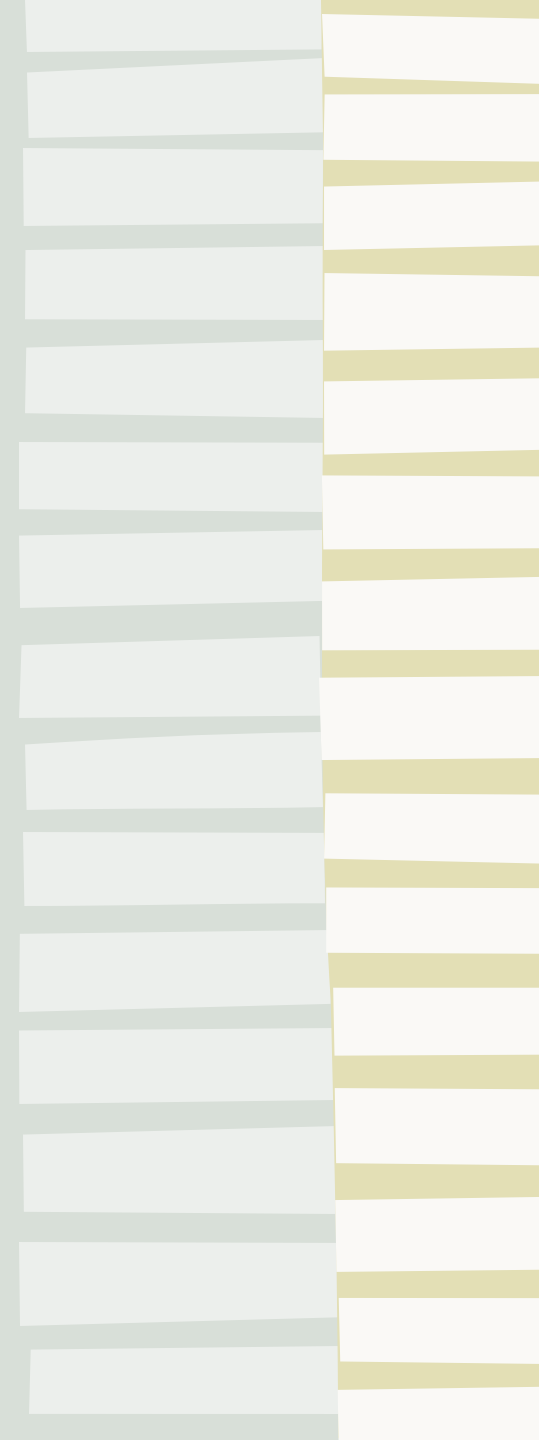
Le script vérifie l'existence d'un groupe et le crée avec groupadd s'il n'existe pas déjà.

## Création d'utilisateurs et assignation

Les utilisateurs sont créés avec `useradd` et peuvent être ajoutés à un groupe avec `usermod -aG`.



# Algorigramme de Création



# Algorigramme de création/Ajout d'utilisateur

PRENONS LE CAS DU CHOIX 'U' (CREATION UTILISATEUR)





## Début de l'algorithme – choix de l'utilisateur du programme

- L'algorithme commence par un menu principal où l'utilisateur choisit l'action à effectuer : créer un utilisateur en entrant u comme utilisateur,
- créer un groupe en entrant g comme 'groupe',
- ou terminer le programme entrant f comme 'fin'.
- Ce choix détermine immédiatement la branche du processus qui sera exécutée.

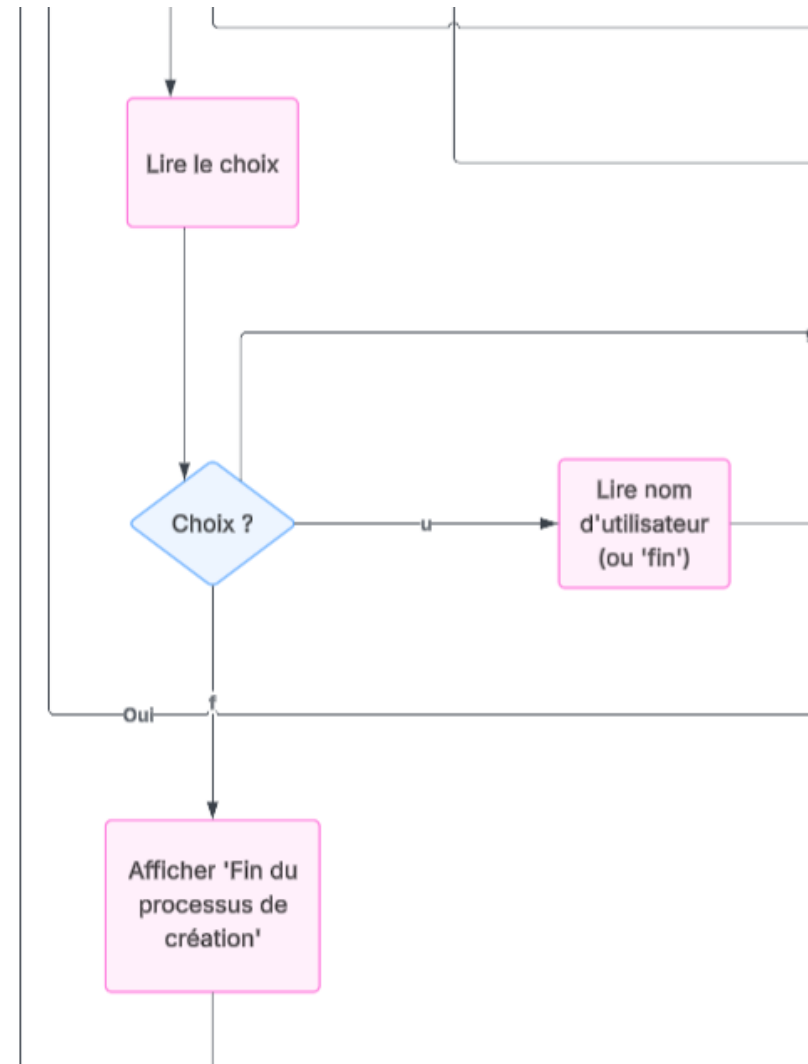
# Lecture du nom d'utilisateur

Le script demande :

« **Nom de l'utilisateur à créer (ou 'fin' pour revenir au menu)** »

Ici, le mot “**fin**” déclenche une **fin locale** :

➡ on **ne quitte pas le programme**, on **revient au menu principal**.



# Vérification de la validité du nom

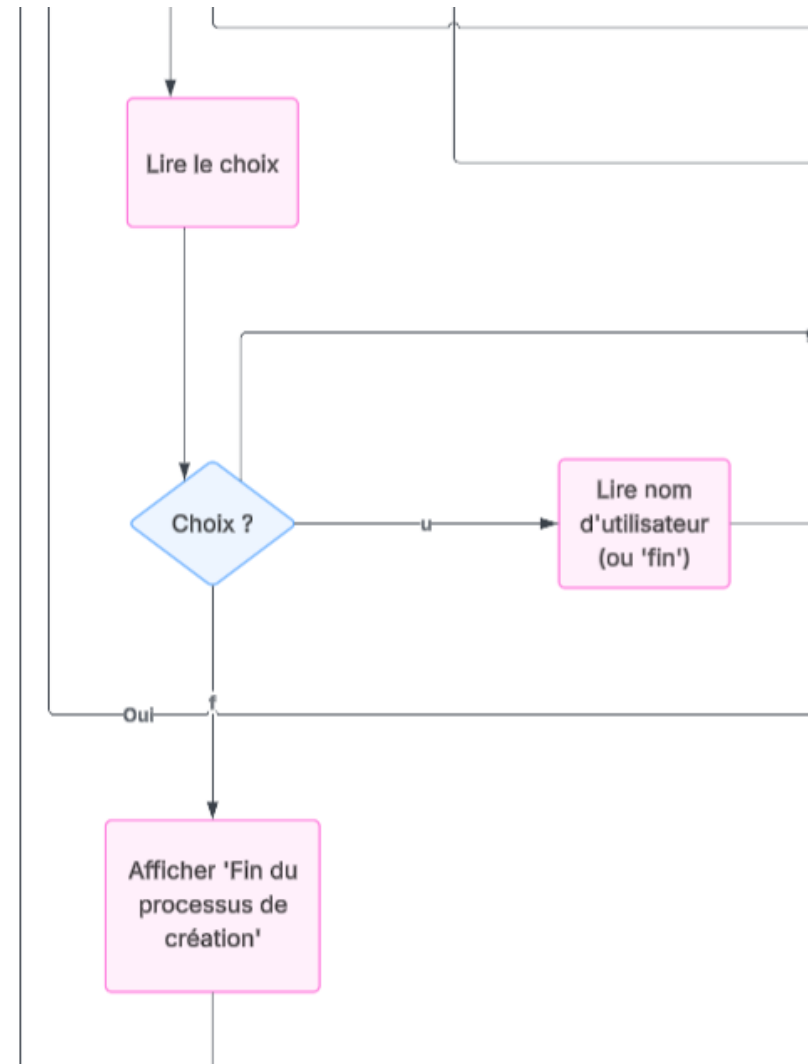
Le script vérifie si le nom suit les règles :

- lettres,
- chiffres,
- tirets,
- underscores.

Si le nom est invalide :

- un message d'erreur s'affiche,
- aucune création n'est effectuée,
- et **retour direct au menu principal**.

C'est un second cas de **fin locale**.



# Vérification de l'existence de l'utilisateur

Le script vérifie ensuite s'il existe déjà.

Deux possibilités :

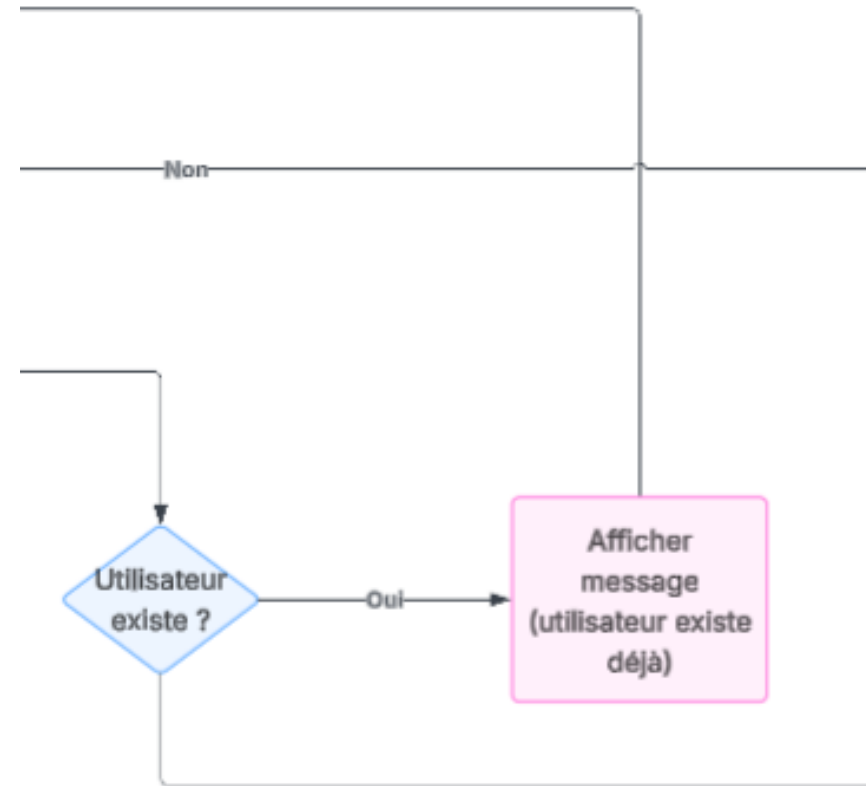
✓ L'utilisateur existe déjà

•Message : "L'utilisateur existe déjà."

•**Retour au menu principal (fin locale)**

✓ L'utilisateur n'existe pas

•Le script peut procéder à la création.



# Création de l'utilisateur

Commande :

```
sudo useradd -m utilisateur
```

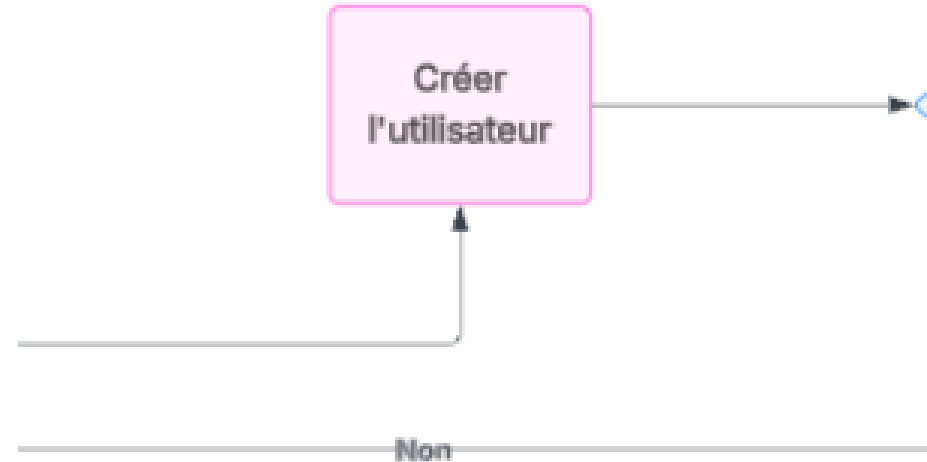
Si la création réussit :

➡ l'utilisateur est créé.

Si elle échoue :

➡ message d'erreur, **retour au menu**.

Encore une **fin locale**.





# Proposition d'ajout à un groupe

Le script demande :

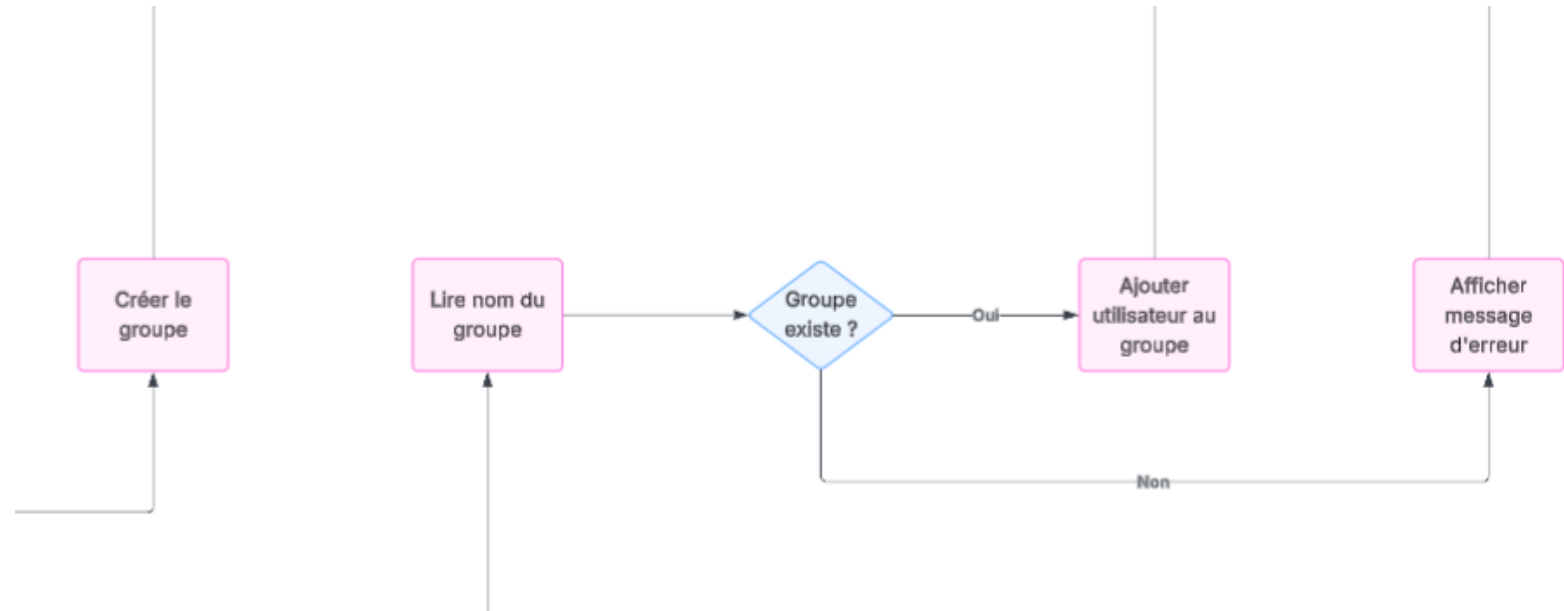
« **Voulez-vous ajouter cet utilisateur à un groupe existant ? (o/n)** »

Deux cas :

- **o** → on demande le nom du groupe.
  - S'il existe → ajout.
  - Sinon → erreur.
- **n** → on passe directement au menu.

Dans tous les cas :

➡ **retour au menu principal** (fin locale).



# Optimisations et évolutions envisageables

## **Interface utilisateur améliorée**

Ajouter une interface colorée et simplifiée pour faciliter l'utilisation, surtout pour les novices.

## **Système de journalisation**

Mettre en place un système de log pour tracer les actions et faciliter le diagnostic des erreurs.

## **Gestion des droits d'accès**

Intégrer chmod et chown pour configurer automatiquement les permissions des fichiers et dossiers.

## **Messages d'erreur explicites**

Rendre les erreurs plus claires en précisant le problème exact et en proposant des solutions.



# Synthèse et perspectives

## Automatisation avec Bash

Le projet a permis de créer des scripts Bash robustes pour automatiser la gestion des utilisateurs et groupes sous Linux Ubuntu.

## Compétences acquises

Les étudiants ont renforcé leurs connaissances des boucles, conditions, tests, et commandes système essentielles à l'administration.

## Perspectives d'amélioration

Le projet ouvre la voie à des améliorations futures telles que interfaces graphiques, journalisation et gestion avancée des permissions.

## Objectifs pédagogiques atteints

Ce TP combine théorie et pratique, préparant les étudiants aux défis professionnels de l'administration Linux.

