

---

# ADMINISTRATION LINUX

## GESTION DES UTILISATEURS

- GESTION DES UTILISATEURS ET GROUPE LOCAUX
- PROFILS ET ENVIRONNEMENTS

# Qu'est-ce qu'un utilisateur ?

- Un compte d'*utilisateur* fournit des limites de sécurité entre différentes personnes et divers programmes pouvant exécuter des commandes.
- Les utilisateurs ont des noms d'utilisateur pour les identifier aux utilisateurs humains et faciliter le travail.
  - En interne, UID
  - Dans la plupart des scénarios, si un humain utilise un compte d'utilisateur, le système attribue un mot de passe secret à l'utilisateur pour prouver qu'il est l'utilisateur autorisé à se connecter.

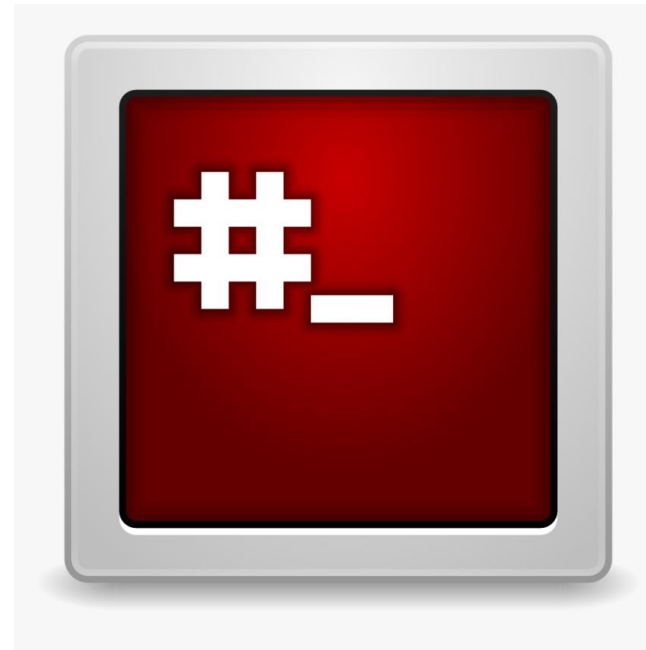
Les trois principaux types de comptes d'utilisateur sont :

- Super utilisateur
- Utilisateur système
- Utilisateur normal



# Le compte super utilisateur

- administre le système et y dispose un accès complet.
- Le nom du super utilisateur est root
- le compte est associé à l'UID 0



# Les comptes utilisateur système

- Sont utilisés par des processus fournissant des services
- Ces processus, ou démons, n'ont généralement pas besoin de s'exécuter en tant que super utilisateur.
- Il s'agit de comptes sans privilège assignés qui permettent de sécuriser leurs fichiers et d'autres ressources les uns des autres et des utilisateurs standard du système
- Ces utilisateurs ne se connectent pas de manière interactive à l'aide d'un compte d'utilisateur système



## Les comptes d'utilisateur standard

- La plupart des utilisateurs ont des comptes d'*utilisateurs standard*
- Utilisé pour leur travail quotidien
- Comme les utilisateurs système, les utilisateurs standard ont un accès limité au système

# Id

- Utilisez la commande id pour afficher des informations sur un utilisateur :

```
[user01@host ~]$ id  
uid=1000(user01) gid=1000(user01) groups=1000(user01)
```

Pour afficher des informations basiques sur un autre utilisateur, transmettez le nom d'utilisateur à la commande id comme argument :

```
[user01@host ~]$ id user02  
uid=1002(user02) gid=1001(user02) groups=1001(user02)
```

- 
- La mise en correspondance des noms d'utilisateur et des UID est définie dans les bases de données des informations sur les comptes.
  - Par défaut, les systèmes utilisent le fichier `/etc/passwd` pour stocker les informations concernant les utilisateurs locaux.
  - Chaque ligne du fichier `/etc/passwd` contient des informations sur un utilisateur. Le fichier est divisé en sept champs séparés par deux-points.
  - Voici un exemple de ligne de `/etc/passwd` :

```
[user01@host ~]$ cat /etc/passwd
```

```
user01:x:1000:1000:User One:/home/user01:/bin/bash
```



- Les 7 blocs sont :

- `user01` : nom d'utilisateur de cet utilisateur.
- `x` : le mot de passe chiffré de l'utilisateur était historiquement stocké ici.
- `1000` : numéro UID de ce compte utilisateur.
- `1000` : numéro GID du groupe principal de ce compte utilisateur.
- `User One` : bref commentaire, description ou nom réel de cet utilisateur.
- `/home/user01` : répertoire personnel de l'utilisateur et répertoire de travail initial au démarrage du shell de connexion.
- `/bin/bash` : programme shell par défaut de cet utilisateur, qui s'exécute lors de la connexion.



## Qu'est-ce qu'un groupe ?

- Un groupe est un ensemble d'utilisateurs
- Partageant l'accès aux fichiers et aux autres ressources du système
- Les groupes peuvent être utilisés pour accorder l'accès à des fichiers à un ensemble d'utilisateurs plutôt qu'à un seul utilisateur.
- Comme les utilisateurs, les groupes ont des noms de groupe pour une reconnaissance plus facile.
- GID définie dans les bases de données de gestion des identités des comptes de groupe `/etc/group`
- Chaque ligne du fichier `/etc/group` contient des informations sur un groupe.
- Chaque entrée de groupe est divisée en quatre champs séparés par deux-points.
- Voici un exemple de ligne de `/etc/group` :

```
[user01@host ~]$ cat /etc/group
```

```
group01:x:10000:user01,user02,user03
```



- Considérez chaque partie du bloc de code, séparée par deux points :
  - group01 : nom de ce groupe.
  - x : champ Mot de passe de groupe obsolète
  - 10000 : numéro GID de ce groupe (10000).
  - user01,user02,user03 : liste des utilisateurs membres de ce groupe en tant que groupe secondaire.

```
[formation@localhost ~]$ cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:formation
cdrom:x:11:
```

## Groupes primaires et groupes secondaires

- Chaque utilisateur a exactement un groupe principal répertorié par GID dans le fichier /etc/passwd.
- Le groupe principal possède les fichiers créés par l'utilisateur.
- Lors de la création d'un utilisateur standard, un groupe est créé avec le même nom que l'utilisateur, pour être le groupe principal de l'utilisateur.
- L'utilisateur est le seul membre de ce groupe privé d'utilisateurs.
- Les utilisateurs peuvent également avoir des groupes secondaires (/etc/group)
- Les utilisateurs ont accès aux fichiers en fonction de l'accès de l'un de leurs groupes indépendamment qu'il s'agisse de groupes principaux ou secondaires.
- Par exemple, si l'utilisateur user01 a un groupe principal user01 et les groupes supplémentaires wheel et webadmin, il peut alors lire les fichiers lisibles par l'un de ces trois groupes.

La commande id peut afficher l'appartenance à un groupe pour un utilisateur

```
[user01@host ~]$ id
```

```
uid=1001(user01) gid=1003(user01) groups=1003(user01),10(wheel),10000(webadmin)
```

## Accès en tant que super utilisateur : root

- Cet utilisateur a le pouvoir d'outrepasser les privilèges normaux sur le système de fichiers
- Utiliser pour gérer et administrer le système.
- Pour les tâches comme l'installation ou la suppression de logiciels, et pour gérer des fichiers et des répertoires du système, l'utilisateur doit augmenter ses privilèges au niveau de ceux de l'utilisateur root.
- En règle générale, seul l'utilisateur root, par rapport aux utilisateurs normaux, peut contrôler la plupart des périphériques, à quelques exceptions près (USB)
- L'utilisateur root a une capacité infinie à endommager le système : suppression de fichiers et de répertoires, suppression de comptes d'utilisateur, ajout de portes dérobées, etc.
- Si le compte d'utilisateur root est compromis, le système est en danger et vous risquez de perdre le contrôle administratif.
- On encourage les administrateurs à se connecter en tant qu'utilisateur normal et à n'augmenter leurs privilèges au niveau root que lorsque cela est nécessaire



# Changement de compte d'utilisateur

- Avec la commande su, les utilisateurs peuvent basculer vers le compte d'un autre utilisateur.
- Demander un mot de passe dépendra si vous exécutez la commande su à partir d'un compte d'utilisateur standard ou root
- Cet exemple utilise la commande su du compte user01 pour basculer vers le compte user02 :

```
[user01@host ~]$ su - user02  
Password: user02_password  
[user02@host ~]$
```

Si vous omettez le nom d'utilisateur, la commande su ou su - tente de basculer vers root par défaut.

```
[user01@host ~]$ su -  
Password: root_password  
[root@host ~]#
```

## Exécution de commandes avec Sudo

- Pour des raisons de sécurité, dans certains cas, les administrateurs système configurent l'utilisateur root pour qu'il n'ait pas de mot de passe valide. ( serveurs critiques...)
- Les utilisateurs ne peuvent pas se connecter au système en tant que root directement avec un mot de passe.
- la commande sudo est la solution pour obtenir l'accès root.
- su VS sudo : Sudo nécessite généralement que les utilisateurs entrent leur propre mot de passe pour l'authentification, et non le mot de passe du compte d'utilisateur auquel ils tentent d'accéder.
- Autrement dit, les utilisateurs qui utilisent la commande sudo pour exécuter des commandes en tant que root n'ont pas besoin de connaître le mot de passe root. Au lieu de cela, ils utilisent leurs propres mots de passe pour authentifier l'accès.

```
[formation@localhost ~]$  
[formation@localhost ~]$ sudo cat /etc/sudoers  
[sudo] Mot de passe de formation :
```

# Aperçu sur la configuration de Sudo

- Le fichier `/etc/sudoers` est le fichier de configuration principal associé à la commande `sudo`.
- la commande `visudo` spéciale vous permettra de modifier ce fichier.
- L'éditeur `visudo` valide également le fichier pour s'assurer qu'il n'y a pas d'erreur de syntaxe.

Par exemple, la ligne suivante du fichier `/etc/sudoers` active l'accès `sudo` pour les membres du groupe `wheel`.

```
%wheel    ALL=(ALL:ALL)    ALL
```

- La chaîne `%wheel` est l'utilisateur ou le groupe auquel la règle s'applique.
- Le symbole `%` avant le mot `wheel` spécifie un groupe.
- La commande `ALL=(ALL:ALL)` spécifie que sur n'importe quel hôte avec ce fichier (le premier `ALL`), les utilisateurs du groupe `wheel` peuvent exécuter des commandes en tant que n'importe quel autre utilisateur (le deuxième `ALL`) et n'importe quel autre groupe (le troisième `ALL`) sur le système.
- La commande `ALL` finale spécifie que les utilisateurs du groupe `wheel` peuvent exécuter n'importe quelle commande.

# Aperçu sur la configuration de Sudo

- L'utilisateur ec2-user peut exécuter une commande en tant que root sans mot de passe:

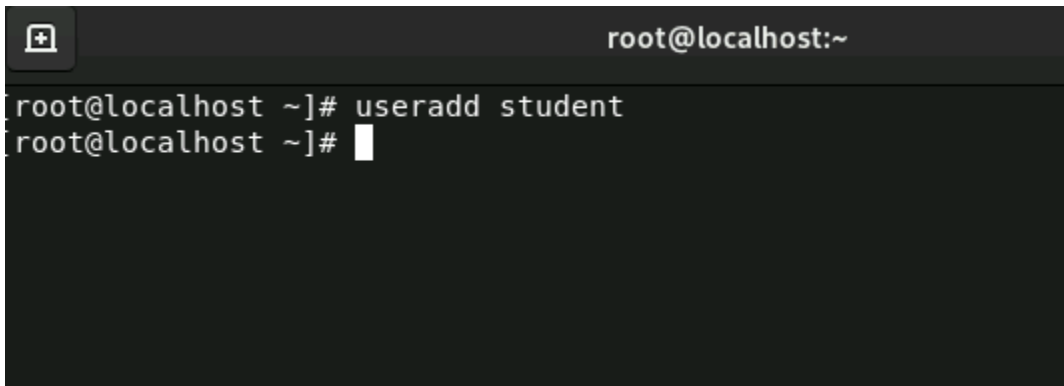
```
ec2-user    ALL=(ALL)    NOPASSWD:ALL
```



# Gestion des comptes d'utilisateur locaux

Création d'utilisateurs à partir de la ligne de commande

- La commande `useradd username` crée un utilisateur appelé `username`.
- Elle configure le répertoire personnel et les informations de compte de l'utilisateur, et crée un groupe privé pour l'utilisateur appelé `username`
- À ce stade, aucun mot de passe valide n'est défini pour le compte et l'utilisateur ne peut pas se connecter avant qu'un mot de passe n'ait été défini.



```
root@localhost:~  
root@localhost ~]# useradd student  
root@localhost ~]#
```

Utiliser les options :

Syntaxe : `useradd [options] nom_utilisateur`

options :

- u pour fixer l'identifiant uid
- p Pour créer un mot de passe (password)
- g Group principal, (gid)
- G groupes secondaires
- c Commentaire
- d le répertoire Personnel ( /home/ )
- s Par défaut, attribution du shell par défaut bash
- e la date d'expiration du compte (format MM/JJ/AA)

## La commande usermod

Options usermod :	Utilisation
-a, --append	À utiliser avec l'option -g pour ajouter les groupes secondaires à l'ensemble de groupes auxquels l'utilisateur appartient au lieu de remplacer l'ensemble de groupes secondaires par un nouvel ensemble.
-c, --comment COMMENT	Ajouter le texte COMMENT au champ de commentaire.
-d, --home HOME_DIR	Spécifier un répertoire personnel pour le compte d'utilisateur.
-g, --gid GROUP	Spécifier le groupe principal du compte d'utilisateur.
-G, --groups GROUPS	Spécifier une liste de groupes secondaires séparés par des virgules pour le compte d'utilisateur.
-L, --lock	Verrouiller le compte.
-m, --move-home	Déplacer le répertoire personnel de l'utilisateur vers un nouvel emplacement. Vous devez l'utiliser avec l'option -d.
-s, --shell SHELL	Spécifier un shell de connexion spécifique pour le compte d'utilisateur.
-U, --unlock	Déverrouiller le compte.

## Suppression d'utilisateurs à partir de la ligne de commande

- La commande `userdel username` supprime l'utilisateur `username` de `/etc/passwd`, mais laisse le répertoire personnel de l'utilisateur intact.
- La commande `userdel -r username` supprime l'utilisateur de `/etc/passwd` ainsi que le répertoire personnel de l'utilisateur.

### AVERTISSEMENT

- Lorsque vous supprimez un utilisateur sans spécifier l'option `userdel -r`, les fichiers de l'utilisateur appartiennent à un UID non affecté.
- Si vous créez un utilisateur et que l'UID de l'utilisateur supprimé est affecté à cet utilisateur, le nouveau compte sera propriétaire de ces fichiers, ce qui représente un risque pour la sécurité.
- En règle générale, les politiques de sécurité de l'organisation interdisent la suppression des comptes d'utilisateur et empêchent leur utilisation, pour éviter ce scénario.

## Définition de mots de passe à partir de la ligne de commande

- La commande `passwd username` définit le mot de passe initial ou modifie le mot de passe existant de l'utilisateur `username`.
- L'utilisateur `root` peut attribuer n'importe quelle valeur à un mot de passe. Le terminal affiche le message si le mot de passe ne satisfait pas aux critères minimaux recommandés, mais vous pouvez retaper le nouveau mot de passe.
- La commande `passwd` le met à jour correctement.

```
[root@localhost ~]# passwd student
Changement de mot de passe pour l'utilisateur student.
Nouveau mot de passe :
MOT DE PASSE INCORRECT : Le mot de passe comporte moins de 8 caractères
Retapez le nouveau mot de passe :
passwd : mise à jour réussie de tous les jetons d'authentification.
[root@localhost ~]#
```

## Plages d'UID

- Linux utilise des numéros et plages de numéros UID à des fins spécifiques.
- UID 0 : UID du compte de super utilisateur (root).
- UID 1-200 : UID de compte système affectés de manière statique aux processus système.
- UID 201-999 : UID attribués aux processus système qui ne possèdent pas de fichiers sur ce système. Les logiciels qui nécessitent un UID sans privilège se voient attribuer dynamiquement un UID à partir de ce pool disponible.
- UID 1000+ : plage d'UID à affecter aux utilisateurs standard sans privilèges.

# Gestion des groupes locaux

## Création de groupes à partir de la ligne de commande

- Utilisez la commande `groupadd` pour créer des groupes.
- Sans option, la commande `groupadd` utilise le premier GID disponible dans la plage spécifiée par les variables `GID_MIN` et `GID_MAX` dans le fichier `/etc/login.defs`.
- Par défaut, la commande attribue une valeur GID supérieure à tout autre GID existant, même si une valeur inférieure devient disponible.

La commande `groupadd` avec l'option `-g` spécifie un GID particulier pour le groupe à utiliser.

```
[root@host ~]# groupadd -g 10000 group01  
[root@host ~]# tail /etc/group  
group01:x:10000:
```

## Modification des groupes existants à partir de la ligne de commande

- La commande `groupmod` change les propriétés d'un groupe existant.
- La commande `groupmod` avec l'option `-n` spécifie un nouveau nom pour le groupe.

```
[root@host ~]# groupmod -n group0022 group02  
[root@host ~]# tail /etc/group  
group0022:x:988:
```

- La commande `groupmod` avec l'option `-g` sert à spécifier un nouveau GID.

```
[root@host ~]# groupmod -g 20000 group0022  
[root@host ~]# tail /etc/group  
group0022:x:20000:
```

## Suppression de groupes à partir de la ligne de commande

```
[root@host ~]# groupdel group0022
```

## Modification de l'appartenance à un groupe à partir de la ligne de commande

- Les membres d'un groupe sont contrôlés par la gestion des utilisateurs.
- Utilisez la commande `usermod -g` pour changer le groupe principal d'un utilisateur.

```
[root@host ~]# id user02
uid=1006(user02) gid=1008(user02) groups=1008(user02)
[root@host ~]# usermod -g group01 user02
[root@host ~]# id user02
uid=1006(user02) gid=10000(group01) groups=10000(group01)
```



- Utilisez la commande `usermod -aG` pour ajouter un utilisateur au groupe secondaire.

```
[root@host ~]# id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03)
[root@host ~]# usermod -aG group01 user03
[root@host ~]# id user03
uid=1007(user03) gid=1009(user03) groups=1009(user03),10000(group01)
```

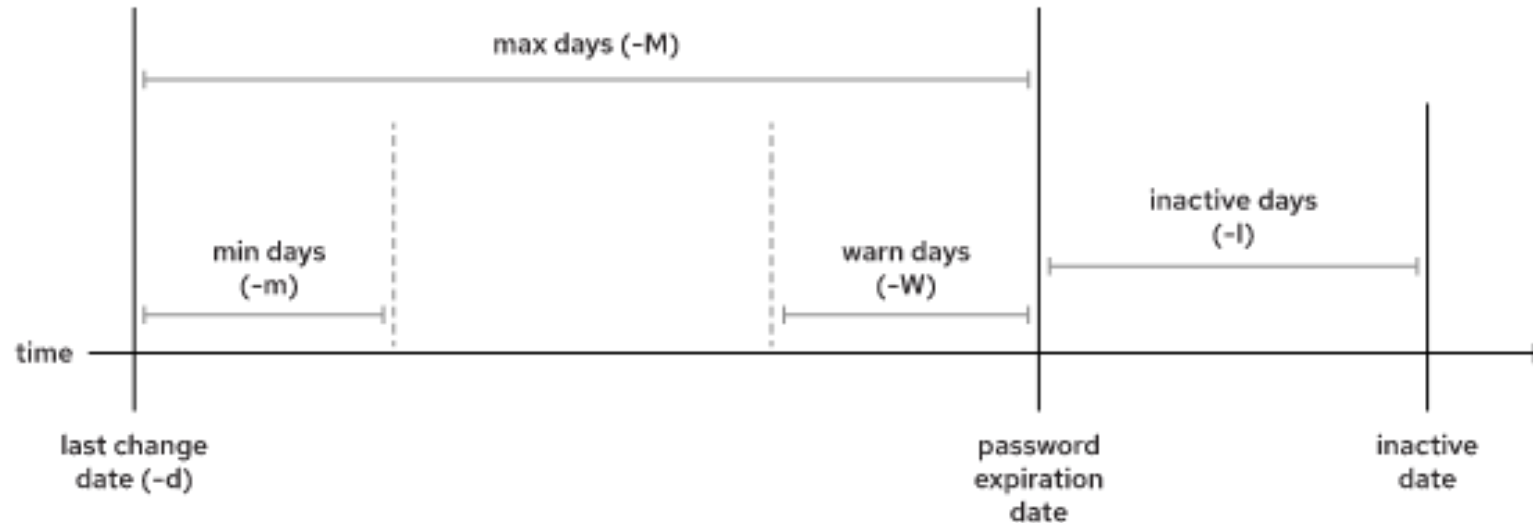
## Modification temporaire de votre groupe principal

- Seul le groupe principal d'un utilisateur est utilisé pour les attributs de création de fichiers.
- Cependant, vous pouvez basculer temporairement votre groupe principal vers un autre groupe, mais vous ne pouvez choisir que parmi les groupes secondaires auxquels vous appartenez déjà.
- Contexte : créer un certain nombre de fichiers, manuellement ou par script, et que vous souhaitez leur attribuer un groupe différent en tant que propriétaire lors de leur création.

```
[user03@host ~]$ newgrp group01
[user03@host ~]# id
uid=1007(user03) gid=10000(group01)
groups=1009(user03),10000(group01)
```

## Configuration du vieillissement du mot de passe

Le diagramme suivant montre les paramètres relatifs au vieillissement d'un mot de passe qui peuvent être ajustés avec la commande chage pour mettre en œuvre une politique de vieillissement de mot de passe.



## Configuration du vieillissement du mot de passe : Exemple

- La commande définit pour le mot de passe de l'utilisateur sysadmin05 :
  - un âge minimum (-m) de zéro jour
  - un âge maximum (-M) de 90 jours
  - une période d'avertissement (-W) de 7 jours
  - une période d'inactivité (-I) de 14 jours

```
[root@host ~]# chage -m 0 -M 90 -W 7 -I 14 sysadmin05
```

## Environnement

Lorsque l'utilisateur se connecte au système, plusieurs fichiers sont lus au lancement du shell pour définir l'environnement de travail :

- /etc/profile
- ~/.bash\_profile, ~/.bash\_login, ~/.profile
- ~/.bashrc

## /etc/profile

- C'est un script shell qui est exécuté en premier lors de la connexion à un terminal texte.
- Ce fichier contient les variables d'environnement de base de tous les processus, et seul l'administrateur système peut le modifier.
- En outre, ce fichier exécute des commandes dans l'environnement du shell de connexion.

Ce script n'est interprété qu'à la connexion de l'utilisateur.

---

`~/.bash_profile`, `~/.bash_login`, `~/.profile`

- Après lecture du fichier `/etc/profile`, Bash recherche le fichier `~/.bash_profile`, `~/.bash_login` ou `~/.profile` dans cet ordre et exécute les commandes contenues dans le premier de ces scripts trouvé et accessible en lecture.
- Ce fichier a la même fonction que le fichier `/etc/profile`, à la différence près qu'il peut être modifié par l'utilisateur pour changer son propre environnement.
- Comme le fichier précédent, ce script n'est interprété qu'à la connexion ; les modifications apportées ne sont prises en compte qu'après reconnexion de l'utilisateur.

## ~/.bashrc

- Le fichier .bashrc permet à chaque utilisateur de personnaliser son shell .
- Ce fichier est cependant réservé aux non-login shells (lancement d'une xterm par exemple).
- C'est le fichier .bash\_profile qui est pris en compte lors du login.

- Ce fichier stocke des alias et des fonctions spécifiques à l'utilisateur.

```
information@localhost ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature
# export SYSTEMD_PAGER=

# User specific aliases and functions
if [ -d ~/.bashrc.d ]; then
    for rc in ~/.bashrc.d/*; do
        if [ -f "$rc" ]; then
            . "$rc"
        fi
    done
fi
```