Linux et Bash - Utilisateurs et droits

45 minutes Normal

DataScientest • com

Introduction à Linux

Utilisateurs et droits

Linux est un système d'exploitation multi-utilisateurs auquel de nombreux utilisateurs peuvent accéder simultanément. Linux peut également être utilisé dans les ordinateurs centraux et les serveurs sans aucune modification (c'est souvent le cas pour la plupart des serveurs Web, où chaque site appartient à un utilisateur. Chaque utilisateur peut se connecter à distance pour maintenir son site).

Mais cela pose des problèmes de sécurité car un utilisateur non sollicité ou malveillant peut corrompre, modifier ou supprimer des données cruciales. Pour une sécurité efficace, Linux divise l'autorisation en 2 niveaux : **La propriété** (utilisateurs) et **les permissions** (droits). Le concept de permission et de propriété des fichiers Linux est crucial dans Linux. Ici, nous allons expliquer ces niveaux. Commençons par la propriété.

Propriété

Utilisateur

Un utilisateur est le propriétaire d'un fichier. Par défaut, la personne qui a créé un fichier en devient le propriétaire. Par conséquent, un utilisateur est aussi parfois appelé propriétaire.

Groupe

Un groupe d'utilisateurs peut contenir plusieurs utilisateurs. Tous les utilisateurs appartenant à un groupe auront le même accès au fichier avec les autorisations du groupe Linux. Supposons que vous ayez un projet dans lequel un certain nombre de personnes doivent avoir accès à un fichier. Au lieu d'attribuer manuellement des autorisations à chaque utilisateur, vous pouvez ajouter tous les utilisateurs à un groupe et attribuer des autorisations de groupe aux fichiers de telle sorte que seuls les membres de ce groupe et personne d'autre ne puisse lire ou modifier les fichiers.

Autres

Machine status

Ubuntu Server 18.04 LTS SSD Volume Type 64-bit x86

Online

34.245.135.23

Connect

Reset Stop

l'autorisation pour les autres, on parle également d'autorisation pour le monde entier.

Super utilisateur et sudo

Nous avons pu voir que certaines permissions nous sont parfois refusées. Pour acquérir à nouveau ces droits, nous nous connectons en tant que super-utilisateur, aussi appelé `root`. Nous pouvons utiliser ce rôle de super-utilisateur de plusieurs façons différentes :

> Connectez vous en tant que super-utilisateur en utilisant la commande `sudo su`.

Vous remarquerez que l'invite de commande a changé : l'habituel `$` a été changé en `#`. D'ailleurs, nous remarquerons que la commande `cd` ne ramène plus à `/home/ubuntu` mais à `/root` . Nous sortons de ce mode super-utilisateur en utilisant la commande `exit`. Si nous utilisons les droits du super-utilisateur pour une seule commande, nous pourrons utiliser la clause `sudo` pour introduire la commande. Ici, notre utilisateur ubuntu n'a pas de mot de passe mais cette commande devrait générer une invite de mot de passe.

Permissions

Chaque fichier et dossier de votre système a les 3 permissions suivantes définies pour les 3 propriétaires mentionnés ci-dessus.

Lire

Cette autorisation vous donne le droit d'ouvrir et de lire un fichier. L'autorisation de lecture sur un répertoire vous donne la possibilité d'énumérer son contenu.

Écriture

Le droit d'écriture vous donne l'autorité de modifier le contenu d'un fichier. L'autorisation d'écriture sur un répertoire vous donne le pouvoir d'ajouter, de supprimer et de renommer les fichiers stockés dans le répertoire. Imaginez que vous avez le droit d'écrire sur un fichier mais pas sur le répertoire dans lequel le fichier est stocké. Vous serez en mesure de modifier le contenu du fichier. Mais vous ne pourrez pas renommer, déplacer ou supprimer le fichier du répertoire.

Exécuter

Vous ne pouvez pas exécuter un programme si la permission d'exécuter n'est pas définie. Si vous n'avez pas les droits d'exécution, vous pouvez toujours voir/modifier le code du programme (à condition que les autorisations de lecture et d'écriture soient définies), mais pas l'exécuter.

Passons à la pratique

Maintenant que nous en savons plus sur la manière dont les fichiers et dossiers sont gérables et par qui ils le sont, nous allons voir dans la pratique comment gérer ces fonctionnalités.

> Listez les éléments disponibles à la racine `/` en utilisant l'argument `-l` :

```
ls -l /
```

Les sorties devraient ressembler à ça :

```
-rw-rw-r-- 1 ubuntu ubuntu 133 Apr 12 11:10 file
```

On peut voir différentes parties.

- `ubuntu` : nom du groupe d'utilisateur du propriétaire ;
- `133` : taille de l'objet ;
- `Apr 12 11:10` : dernière date de modification ;
- `file` : nom de l'objet.

La première partie indique les droits sur l'objet. Nous retrouvons le premier caractère et ensuite 3 chaînes de trois caractères.

- `-` : Cette première partie correspond à la nature de l'objet: `d` pour un dossier, `-` pour un fichier, `l` pour un lien... ;
- `rw-` : correspond aux permissions accordées au propriétaire de l'objet ;
- `rw-` : correspond aux permissions accordées aux utilisateurs qui appartiennent au même groupe que le propriétaire de l'objet ;
- `r--` : correspond aux permissions accordées aux autres utilisateurs.

Ces permissions semblent un peu compliquées mais en réalité, les lettres indiquent que l'autorisation est donnée alors que `-` indique que la permission n'est pas accordée. Ainsi, on peut avoir les lettres suivantes :

- droits en lecture : `r` ;
- droits en écriture : `w` ;
- droits en exécution : `x` ;
- information concernant le fait que ce soit un dossier ou non : `d` .

L'utilisation de groupe d'utilisateurs est intéressante car elle permet de donner des droits à un groupe de personne d'un coup.

Changement de permission

Pour changer les permissions d'un fichier, il faut utiliser la commande `chmod` .

```
1  chmod 777 file
2  chmod a+rwx file
3
```

La première façon consiste à utiliser une représentation binaire des permissions : chaque chiffre correspond à un groupe d'utilisateurs comme vu précédemment lors de la lecture des droits.

- `0` : `---`
- `1` : `--x`
- `2` : `-w-`
- `3` : `-wx`
- `4` : `r--`
- `5` : `r-x`
- `6` : `rw-`
- `7` : `rwx`

La deuxième façon consiste à désigner le ou les groupes à qui l'on veut attribuer ou enlever des droits :

- `a` : les utilisateurs concernés par la modification `a` pour tous (`all`), `u` pour le propriétaire, `g` pour le groupe de l'utilisateur `o` pour les autres utilisateurs ;
- `+` : est ce qu'on donne ou enlève des droits: `+` pour donner et `-` enlever ;
- `rwx` : Les droits à ajouter ou retirer.

Notez enfin que pour modifier les droits d'un fichier, il faut en être le propriétaire ou utiliser le super-utilisateur.

Validated

Robin
BIRON