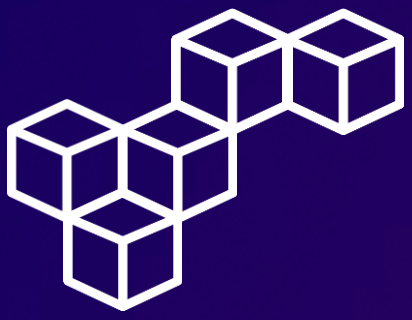
AWS re:Start LAB

Managing File Permissions

WEEK 2

aws re/start



Overview

Managing file permissions is essential for system security and access control. It involves setting rules to determine who can access, modify, or execute files and directories, preventing unauthorized access and data breaches. Assigning ownership and group settings further refines access control, promoting system stability and reliability by minimizing conflicts and ensuring that files are accessed only by authorized entities.

Proper file permission management enhances system organization and reduces the risk of data loss or corruption. Administrators can use commands to adjust permissions accurately, maintaining a secure and well-structured file system. This proactive approach to permission management not only safeguards sensitive information but also contributes to overall system performance and reliability, making it a fundamental practice in system administration.

Note: This lab was made using Windows Subsystem for Linux.

Topics covered

- Change folder and file ownership to match the appropriate company structure
- Change permission modes
- Modify file permissions for a user
- Assign appropriate permissions

Task 1

Use SSH to connect to an Amazon Linux EC2 instance

Initial Preparations

In the AWS Management Console, select the EC2 instance and make note of the **Public IPv4 address**.

Download the **private key file** labsuser.pem. Change to the Downloads directory and modify the permissions on the key to be read-only (r--------).

Connect to the instance using SSH

Establish a connection to the EC2 instance using the ssh command, the key and the instance's public IPv4 address.

```
support@HP-Pavilion-Laptop:~/Downloads$ ssh -i labsuser.pem ec2-user@54.185.134.90
The authenticity of host '54.185.134.90 (54.185.134.90)' can't be established.
ED25519 key fingerprint is SHA256:T3YM8YyWYizJBXzR/7uhrEteutwUmaUSnSor2NAVh20.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.185.134.90' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-10-0-10-208 ~]$
```

aws re/start

Task 2

Change file and folder ownership

Folder ownership

Using the recursive chown –R command change:

- The **companyA** folder ownership to mjackson, the CEO, and group ownership to Personnel.
- The **HR** folder ownership to ljuan, the HR manager, and group ownership to HR.
- The **Finance** folder ownership to mmajor, the finance manager, and group ownership to Finance.

```
[ec2-user@ip-10-0-10-208 ~]$ pwd
/home/ec2-user
[ec2-user@ip-10-0-10-208 ~]$ cd companyA
[ec2-user@ip-10-0-10-208 companyA]$ sudo chown -R mjackson:Personnel /home/ec2-user/companyA
[ec2-user@ip-10-0-10-208 companyA]$ sudo chown -R ljuan:HR HR
[ec2-user@ip-10-0-10-208 companyA]$ sudo chown -R mmajor:Finance HR/Finance
[ec2-user@ip-10-0-10-208 companyA]$
```

Review ownership

Validate your work by using the ls –laR command. I will display the output the ls –ld command for clearer visibility. Notice that after making the previous ownership changes, the ec2-user no longer has write permissions for the **companyA** directory.

```
[ec2-user@ip-10-0-10-208 companyA]$ ls -ld /home/ec2-user/companyA HR HR/Finance
drwxr-xr-x 10 mjackson Personnel 147 Apr  8 20:43 /home/ec2-user/companyA
drwxr-xr-x  6 ljuan    HR         72 Apr  8 20:43 HR
drwxr-xr-x  2 mmajor   Finance   105 Apr  8 20:43 HR/Finance
[ec2-user@ip-10-0-10-208 companyA]$
```

Task 3

Change permission modes

The symbolic and absolute modes

Enable write permissions for others in the **companyA** folder. Afterward, create a couple of files and demonstrate the use of the symbolic mode and the absolute mode with the chmod command to change the file permissions.

```
[ec2-user@ip-10-0-10-208 companyA]$ sudo chmod o+w /home/ec2-user/companyA
[ec2-user@ip-10-0-10-208 companyA]$ vi symbolic_mode_file
[ec2-user@ip-10-0-10-208 companyA]$ chmod g+w symbolic_mode_file
[ec2-user@ip-10-0-10-208 companyA]$ vi absolute_mode_file
[ec2-user@ip-10-0-10-208 companyA]$ chmod 764 absolute_mode_file
[ec2-user@ip-10-0-10-208 companyA]$
```

Review file permissions

Use the ls-l command to observe the changes you made. Notice that the **absolute_mode_file** is now displayed in green color, indicating that it has become an executable file due to the permissions granted.

```
[ec2-user@ip-10-0-10-208 companyA]$ ls -l
total 0
drwxr-xr-x 2 mjackson Personnel  6 Apr  8 20:43 CEO
drwxr-xr-x 2 mjackson Personnel  6 Apr  8 20:43 Documents
drwxr-xr-x 2 mjackson Personnel  6 Apr  8 20:43 Employees
drwxr-xr-x 6 ljuan    HR        72 Apr  8 20:43 HR
drwxr-xr-x 2 mjackson Personnel  6 Apr  8 20:43 Management
-rw-r--r-- 1 mjackson Personnel  0 Apr  8 20:43 Roster.csv
drwxr-xr-x 2 mjackson Personnel  6 Apr  8 20:43 Sales
drwxr-xr-x 2 mjackson Personnel 24 Apr  8 20:43 SharedFolders
drwxr-xr-x 2 mjackson Personnel  6 Apr  8 20:43 Shipping
-rwxrw-r-- 1 ec2-user ec2-user   0 Apr  8 20:57 absolute_mode_file
-rw-rw-r-- 1 ec2-user ec2-user   0 Apr  8 20:57 symbolic_mode_file
[ec2-user@ip-10-0-10-208 companyA]$
```

Task 4

Assign permissions

The Shipping folder

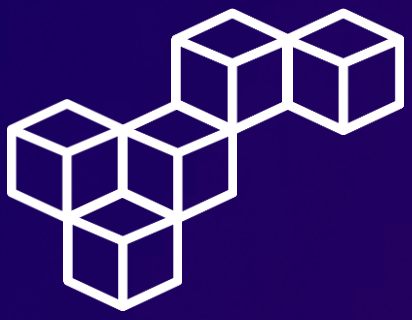
Using the chown -R command, change the ownership of the **Shipping** folder to eowusu, the current shipping manager, and the group ownership to Shipping.

```
[ec2-user@ip-10-0-10-208 companyA]$ pwd
/home/ec2-user/companyA
[ec2-user@ip-10-0-10-208 companyA]$ sudo chown -R eowusu:Shipping Shipping
[ec2-user@ip-10-0-10-208 companyA]$ ls -laR Shipping
Shipping:
total 0
drwxr-xr-x  2 eowusu   Shipping    6 Apr  8 20:43 .
drwxr-xrwx 10 mjackson Personnel 199 Apr  8 20:57 ..
[ec2-user@ip-10-0-10-208 companyA]$
```

The Sales folder

Using the chown -R command, also change the ownership of the **Sales** folder to nwolf, the current sales manager, and the group ownership to Sales.

```
[ec2-user@ip-10-0-10-208 companyA]$ sudo chown -R nwolf:Sales Sales
[ec2-user@ip-10-0-10-208 companyA]$ ls -laR Sales
Sales:
total 0
drwxr-xr-x  2 nwolf    Sales       6 Apr  8 20:43 .
drwxr-xrwx 10 mjackson Personnel 199 Apr  8 20:57 ..
[ec2-user@ip-10-0-10-208 companyA]$
```



Conclusions

The chown command

Managing file and folder ownership plays a pivotal role in data security and accountability. By assigning ownership to specific users or groups, administrators can track actions on files and directories, ensuring that only authorized individuals have access. This practice helps maintain data integrity and security, fostering effective collaboration and preventing unauthorized modifications.

The chmod command

Managing permission modes is essential for regulating access based on user roles. By setting permission modes like read, write, and execute for owners, groups, and others, administrators enforce security policies and prevent unauthorized access. This level of control ensures data protection and efficient resource utilization, contributing significantly to overall system security and reliability.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru