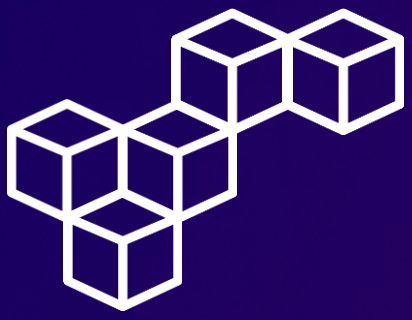
AWS re:Start LAB

Managing Users and Groups

WEEK 2

aws re/start



Overview

Managing users and groups in Unix-like systems is a fundamental responsibility of system administrators. This task involves creating, modifying, and deleting user accounts and groups to control access to system resources and maintain security. By effectively managing users and groups, administrators can enforce security policies, allocate resources efficiently, and streamline access control mechanisms within the system.

The management of users and groups also includes activities such as assigning appropriate permissions, setting passwords, managing group memberships, and delegating administrative privileges. It is essential for administrators to monitor user activities, ensure compliance with organizational policies, and implement security measures to protect sensitive data. Overall, effective management of users and groups is crucial for maintaining a secure and well-organized computing environment.

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

Topics covered

- Create new users with a default password
- Create groups and assign the appropriate users
- Log in as different users

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@52.43.72.66
The authenticity of host '52.43.72.66 (52.43.72.66)' can't be established.
ED25519 key fingerprint is SHA256:W/Hbe10A2lJF/jUrR1M3aBrfGp6GaJxW3PADVlxjLz4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.43.72.66' (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-10-0-10-108 ~]$
```

aws re/start

Task 2

Create Users

Step 1: Add new users

Use the sudo useradd <username> command to create new users.

```
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd arosalez
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd eowusu
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd jdoe
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd ljuan
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd mmajor
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd mjackson
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd nwolf
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd psantos
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd smartinez
[ec2-user@ip-10-0-10-108 ~]$ sudo useradd ssarkar
[ec2-user@ip-10-0-10-108 ~]$
```

Step 2: Set user passwords

Use the sudo passwd <username> command to establish user passwords.

```
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd arosalez
Changing password for user arosalez.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd eowusu
Changing password for user eowusu.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd jdoe
Changing password for user jdoe.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd ljuan
Changing password for user ljuan.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd mmajor
Changing password for user mmajor.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Task 2

Create Users

Continue using the sudo passwd <username> command to establish the remaining user passwords.

```
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd mjackson
Changing password for user mjackson.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd nwolf
Changing password for user nwolf.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd psantos
Changing password for user psantos.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd smartinez
Changing password for user smartinez.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$ sudo passwd ssarkar
Changing password for user ssarkar.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-10-0-10-108 ~]$
```

Step 3: Check user creation

To validate that all users have been created, enter the command sudo cat /etc/passwd | cut -d: -f1.

```
[ec2-user@ip-10-0-10-108 ~]$ sudo cat /etc/passwd | cut -d: -f1 | tail -n 11
ec2-user
arosalez
eowusu
jdoe
ljuan
mmajor
mjackson
nwolf
psantos
smartinez
ssarkar
[ec2-user@ip-10-0-10-108 ~]$
```

Task 3

Create Groups

Step 1: Add new groups

Use the sudo groupadd <group name> command to create new groups. In this lab we will not establish passwords for the group accounts.

```
[ec2-user@ip-10-0-10-108 ~]$ sudo groupadd Sales
[ec2-user@ip-10-0-10-108 ~]$ sudo groupadd HR
[ec2-user@ip-10-0-10-108 ~]$ sudo groupadd Finance
[ec2-user@ip-10-0-10-108 ~]$ sudo groupadd Shipping
[ec2-user@ip-10-0-10-108 ~]$ sudo groupadd Managers
[ec2-user@ip-10-0-10-108 ~]$ sudo groupadd CEO
[ec2-user@ip-10-0-10-108 ~]$
```

Step 2: Check group creation

To verify that all the groups were added, enter the command cat /etc/group. Notice that the new groups were assigned a group identifier (GID) but do not have any members yet.

```
[ec2-user@ip-10-0-10-108 ~]$ cat /etc/group | tail -n 6
Sales:x:1011:
HR:x:1012:
Finance:x:1013:
Shipping:x:1014:
Managers:x:1015:
CEO:x:1016:
[ec2-user@ip-10-0-10-108 ~]$
```

Task 3

Create Groups

Step 3: Add users to groups

Use the sudo usermod -a -G <group name> <username> command to add all the users to the appropriate groups.

```
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Sales,Managers arosalez
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Shipping eowusu
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Shipping jdoe
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G HR,Managers ljuan
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Finance,Managers mmajor
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G CEO mjackson
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Sales nwolf
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Shipping psantos
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G HR smartinez
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Finance ssarkar
[ec2-user@ip-10-0-10-108 ~]$ sudo usermod -a -G Sales,HR,Finance,Shipping,Managers,CEO ec2-user
[ec2-user@ip-10-0-10-108 ~]$
```

Step 4: Check group memberships

To check all the group memberships, enter the command cat /etc/group again. Notice that all the groups now have at least a couple of members.

```
[ec2-user@ip-10-0-10-108 ~]$ cat /etc/group | tail -n 6
Sales:x:1011:arosalez,nwolf,ec2-user
HR:x:1012:ljuan,smartinez,ec2-user
Finance:x:1013:mmajor,ssarkar,ec2-user
Shipping:x:1014:eowusu,jdoe,psantos,ec2-user
Managers:x:1015:arosalez,ljuan,mmajor,ec2-user
CEO:x:1016:mjackson,ec2-user
[ec2-user@ip-10-0-10-108 ~]$
```

Task 4

Log in using the new users

Log in as another user

Enter the command su arosalez and type the respective user password to log in as the user arosalez. Notice that the bash prompt has changed.

```
[ec2-user@ip-10-0-10-108 ~]$ su arosalez
Password:
[arosalez@ip-10-0-10-108 ec2-user]$
```

Write a file to another user's home directory

Enter the pwd command to confirm that we still are in the **/home/ec2-user** directory. If we attempt to create a file with the touch command in this directory, we will receive a Permission denied message because the current logged-in user, arosalez, does not have permission to write files to the ec2-user home directory.

```
[arosalez@ip-10-0-10-108 ec2-user]$ pwd
/home/ec2-user
[arosalez@ip-10-0-10-108 ec2-user]$ touch myFile.txt
touch: cannot touch 'myFile.txt': Permission denied
```

Task 4

Log in using the new users

Force the write with the sudo command

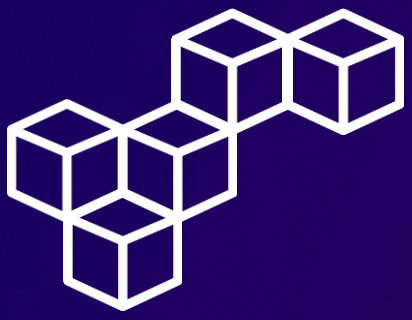
If we attempt to use sudo privilege elevation to write a file to the ec2-user home directory we will receive a message stating that the current logged-in user, arosalez, is not in the sudoers file, and the sudo attempt will be reported to a log file.

```
[arosalez@ip-10-0-10-108 ec2-user]$ sudo touch myFile.txt

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for arosalez:
arosalez is not in the sudoers file.  This incident will be reported.
```

Review sudo logs

Finish the **su session** with the exit command. Then, enter the command sudo cat /var/log/secure to display the content of the secure file. At the end of this file you'll see how a sudo not permitted action was logged into the /var/log/secure file.

```
[arosalez@ip-10-0-10-108 ec2-user]$ exit
exit
[ec2-user@ip-10-0-10-108 ~]$ sudo cat /var/log/secure | grep arosalez | tail -n 3
Apr  7 20:01:38 ip-10-0-10-108 su: pam_unix(su:session): session opened for user arosalez by ec2-user(uid=1000)
Apr  7 20:03:05 ip-10-0-10-108 sudo: arosalez : user NOT in sudoers ; TTY=pts/0 ; PWD=/home/ec2-user ; USER=root ;
 COMMAND=/bin/touch#040myFile.txt
Apr  7 20:03:45 ip-10-0-10-108 su: pam_unix(su:session): session closed for user arosalez
[ec2-user@ip-10-0-10-108 ~]$
```



Conclusions

Creating user accounts

User accounts serve as individual entities through which access permissions and privileges are granted or restricted. By properly managing user accounts, administrators can ensure that users have the necessary access levels to perform their tasks while maintaining system security.

Managing group memberships

Group memberships significantly facilitate administration by allowing administrators to apply permissions and access controls collectively to multiple users. This simplifies the process of managing access rights, as changes made to group memberships automatically apply to all members.

Elevating permissions to administrative privileges

Administrative privileges, such as those granted through sudo and su, are powerful but must be used cautiously to avoid security risks and system integrity compromise. Administrators must follow best practices and limit the use of elevated privileges to authorized tasks only.

Reviewing sudo logs

Sudo logs play a vital role in ensuring accountability and transparency in system administration. They provide a detailed record of commands executed with elevated privileges, including information about who executed the command and when.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru