aws re/start

Users and Groups

Linux Fundamentals

Welcome to Users and Groups.

What you will learn

At the core of the lesson

You will learn how to:

- Manage user accounts
- Manage group accounts
- Elevate permissions by using the `su` and `sudo` commands
- Describe AWS Identity and Access Management (IAM), the authentication service that Amazon Web Services (AWS) uses

aws re/start

In this lesson, you will learn how to:

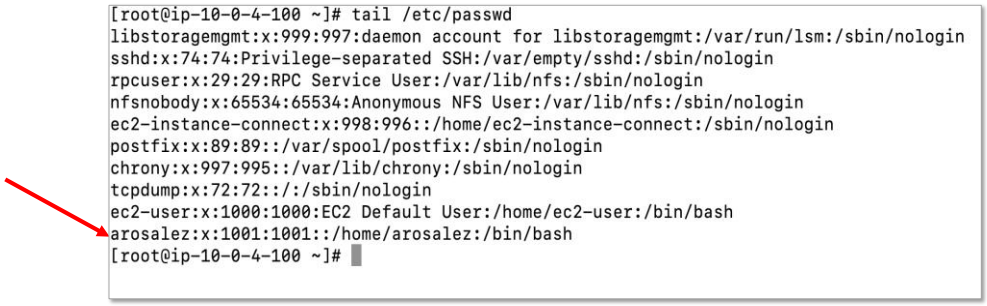
- Manage user accounts
- Manage group accounts
- Elevate permissions by using the `su` and `sudo` commands
- Describe IAM, the authentication service that AWS uses

Managing users

In this section, you will learn how to create users and manage their passwords.

User accounts

- User accounts represent users on the system.
- User information can be stored locally or on another server accessible through a network.
- When information is stored locally, Linux stores it in the **/etc/passwd** file.
- Best practice is to assign one user per account.
- Don't share accounts.



aws re/start

This example shows the content of the `passwd` file.

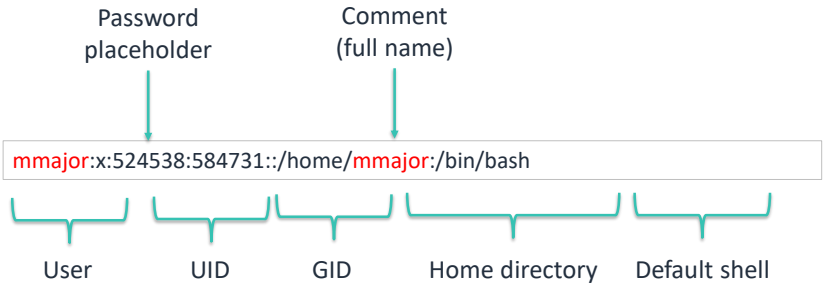
`tail` is a command that displays the last lines of a file. By default, it displays the last 10 lines, but you can adjust the number of lines using the `-n` option.

For example, the following command displays the last five lines:

```
tail -n 5 /etc/passwd
```

The /etc/passwd file

Linux stores the accounts in the /etc/passwd file.



aws re/start

The passwd file contains registered users on the system.

It is formatted as a colon-separated file that contains the following information:

- User name
- Encrypted password
- User ID
- User's group ID
- Full name of the user
- Home directory of the user
- The shell that is used after login

Default user accounts

- Default system accounts are created during the installation of Linux and services.
- For example, a root user account is created during the installation, which allows administration of the system.

```
[root@ip-10-0-4-100 ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```


`head` is the complementary command of tail. It displays the first 10 lines of a file by default.

You can adjust the number of lines by using the `-n` option:

For example, the following command displays the first five lines:

```
head -n 5 /etc/passwd
```

The useradd command

- Creates the user account
- Creates a home directory for the user in /home
- Defines account defaults

```
[root@ip-10-0-4-100 ~]# useradd mmajor
[root@ip-10-0-4-100 ~]# id mmajor
uid=1002(mmajor) gid=1002(mmajor) groups=1002(mmajor)
[root@ip-10-0-4-100 ~]# grep mmajor /etc/passwd
mmajor:x:1002:1002::/home/mmajor:/bin/bash
[root@ip-10-0-4-100 ~]#
```


aws re/start

`grep` is a command that searches for a string in a file.

For example, the following command displays all occurrences of the string `mmajor` in the file `/etc/passwd`:

```
grep mmajor /etc/passwd
```

The `useradd` command options

- Options allow customization of the user account at the time of creation.
- The comment field is often used to hold the user's full name.

Option	Description	Example
-c	Comment	`useradd -c "new employee" jdoe`
-e	Account expiration	`useradd -e 2025-01-01 jdoe`
-d	Home directory path	`useradd -d /users/jdoe jdoe`

This table provides useful `useradd` command options.

The `usermod` command

This command is used to modify or change parts of or a whole existing user account.

Option	Description	Example
`-c`	Comment	`usermod -c "Mary Major" mmajor`
`-e`	Account expiration	`usermod -e 2025-01-01 mmajor`

```
[root@ip-10-0-4-100 ~]# grep mmajor /etc/passwd
mmajor:x:1002:1002::/home/mmajor:/bin/bash
[root@ip-10-0-4-100 ~]# usermod -c "Mary Major" mmajor
[root@ip-10-0-4-100 ~]# grep mmajor /etc/passwd
mmajor:x:1002:1002:Mary Major:/home/mmajor:/bin/bash
```


`grep` is a Linux command to search text.

It can be used as follows to search for the word *hello* in the files that are located in `/etc/passwd`:

```
grep hello /etc/passwd
```

It can also be used with the pipe symbol `(|)` that redirects the output of the command to another command.

- `ls /etc/passwd | grep hello: ls /etc/passwd` lists all the files in the /etc/passwd folder.
- `|` redirects the result to the second command, `grep`, which searches for the word *hello* in the list of files.

The `userdel` command

- Deletes a user account
- Uses the `-r` option to also delete the user's home directory

```
[root@ip-172-31-27-186 ~]# useradd jdoe
[root@ip-172-31-27-186 ~]# id jdoe
uid=1002(jdoe) gid=1002(jdoe) groups=1002(jdoe)
[root@ip-172-31-27-186 ~]# userdel -r jdoe
[root@ip-172-31-27-186 ~]# id jdoe
id: jdoe: no such user
[root@ip-172-31-27-186 ~]#
```


aws re/start

Use the `userdel` command to delete a user account.

The passwd command

- User passwords are set with the passwd command.
- You must enter the password twice.
- Users can reset their own passwords, and the root user can reset any user password.
- No characters are echoed to the screen when the password is set.

```
[root@ip-10-0-4-100 ~]# passwd mmajor
Changing password for user mmajor.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-10-0-4-100 ~]#
```


aws re/start

The passwd command is used to set user passwords.

Managing groups

In this section, you'll learn how to create and manage users and groups.

What are groups?

- A group is a set of accounts.
- Groups are a convenient way to associate user accounts with similar security needs.
- For example, it is easier to grant permissions to a group of four users than to grant permissions individually to each of four users individually.
- The storage location for groups is the **/etc/group** file.

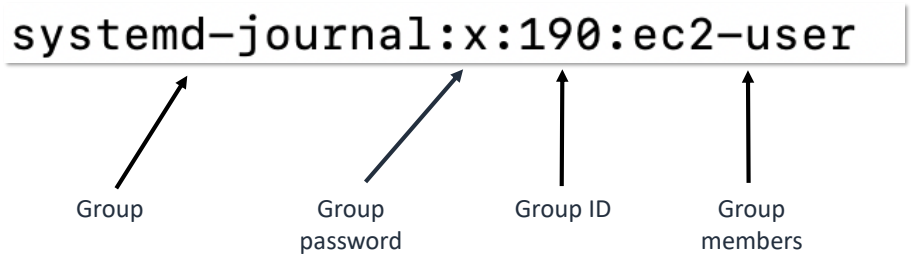
Groups	Users
ec2-user	mmajor, jdoe, ljuan, moliveira
devs	jdoe, wxiulan

aws re/start

For instance, give Read access to a folder or a file to the ec2-user group instead of individually assigning the rights to each user.

The /etc/group file

Storage location for groups



aws re/start

The /etc/group file is as follows:

```
[ec2-user@ip-172-31-27-186 ~]$ tail /etc/group
chrony:x:994:
screen:x:84:
stapusr:x:156:
stapsys:x:157:
stapdev:x:158:
tcpdump:x:72:
ec2-user:x:1000:jdoe
devs:x:1004:jdoe,mmajor
```

The `groupadd`, `groupmod`, and `groupdel` commands

Option	Description	Example
`groupadd`	Creates a new group	`groupadd` *group*
`groupmod`	Modifies an existing group	`groupmod -n` *new_group old_group*
`groupdel`	Deletes an existing group	`groupdel` *group*

```
[root@ip-172-31-27-186 ~]# groupadd marketing
[root@ip-172-31-27-186 ~]# tail -n 3 /etc/group
mmajor:x:1001:
devs:x:1004:mmajor
marketing:x:1005:
```

The `groupadd` command

```
[root@ip-172-31-27-186 ~]# groupdel marketing
[root@ip-172-31-27-186 ~]# tail -n 3 /etc/group
ec2-user:x:1000:
mmajor:x:1001:
devs:x:1004:mmajor
```

The `groupdel` command

aws re/start

Familiarize yourself with the groupadd, groupmod, and groupdel commands.

Add a user to a group

- Adding a user to a group is a modification of a user, not a group.
- To add a user to a group, you can use:
  - The `usermod` command
  - The `gpasswd` command

```
[root@ip-172-31-27-186 ~]# usermod -aG hr,marketing mmajor
[root@ip-172-31-27-186 ~]# gpasswd -a jdoe marketing
Adding user jdoe to group marketing
[root@ip-172-31-27-186 ~]# tail -n 5 /etc/group
mmajor:x:1001:
devs:x:1004:mmajor
jdoe:x:1002:
marketing:x:1005:mmajor,jdoe
hr:x:1006:mmajor
[root@ip-172-31-27-186 ~]#
```


aws re/start

- The `usermod` command adds the user `mmajor` to the groups `hr` and `marketing`:
  `usermod -aG hr,marketing mmajor`
- The gpasswd command adds the user `jdoe` to the `marketing` group:
  `gpasswd -a jdoe marketing`
- To append a user to a group without removing them from other groups, use the `usermod` command with `-aG` options

The `gpassword` command

- Is used to administer the `/etc/group` file
- Usage: `gpasswd [option] GROUP`

Option	Description
`-a, --add`	Add USER to GROUP
`-d, --delete`	Remove USER from GROUP
`-M, --members USER1,USER2,...`	Set the list of members of GROUP
`-A, --administrators ADMIN1,ADMIN2,...`	Set the list of administrators for GROUP

aws re/start

The command `gpasswd -a jdoe ec2-user` adds the user `jdoe` to the `ec2-user` group.

The command `gpasswd –M smartinez,rroe ec2-user` sets the list of members of the `ec2-user` group to `smartinez, rroe.`

User permissions

In this section, you'll learn about user permissions and the usage of the `su` and `sudo` commands to run admin commands.

User permission levels

Root user

- Access any file
- Change any file
- Control services
- Manage any account
- Manage hardware
- Manage the Linux kernel
- Manage software

Standard user

- Access any files if given permissions to do so
- Control any files that the user owns
- Limited access to manage the system

aws re/start

A standard user is a user that only has rights and file permissions that the user's role requires. For instance, a business user might be able to run only certain programs like a document editor and a reporting tool. The user might be able to edit only files that are located in some specific folders.

The root user can run any program and access any file. Only system administrators can use it.

It is important to create a standard user with the right privileges to avoid unauthorized access to files or programs.

Use caution with `root`

- Security best practice: Do not log in to the system with administrative permissions
- Log in as a standard user, and then elevate permissions only when necessary
- The root user is a powerful Linux account; mistakes can make the system inoperable
- The root user command prompt ends with #
- The standard user command prompt ends with $

```
[root@server00 ~]# exit
logout
[userA@server00 ~]$
```

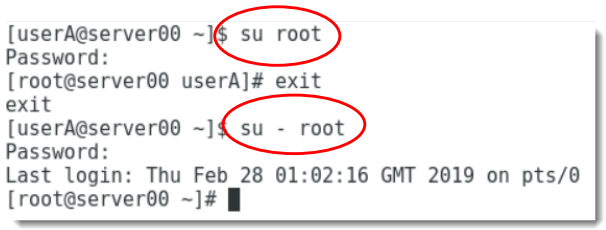

aws re/start

A user with root privileges could mistakenly run a command such as `delete /` that would erase the entire system.

The `su` command

- Log in as a standard user, and then elevate permissions to accomplish administrative tasks.
- Be careful to exit the root context.

Command	Description
`su root`	Switches to root with the current user's environment
`su - root`	Switches to root with the root's environment



aws re/start

`su -` is equivalent to `su - root`.

Only a system administrator should be authorized to run the command `su root`.

`su` stands for substitute user, and you can use `su` to log in as any user, not only the root user. For example, the following command switches the current user to user `student02`:

```
su student02
```

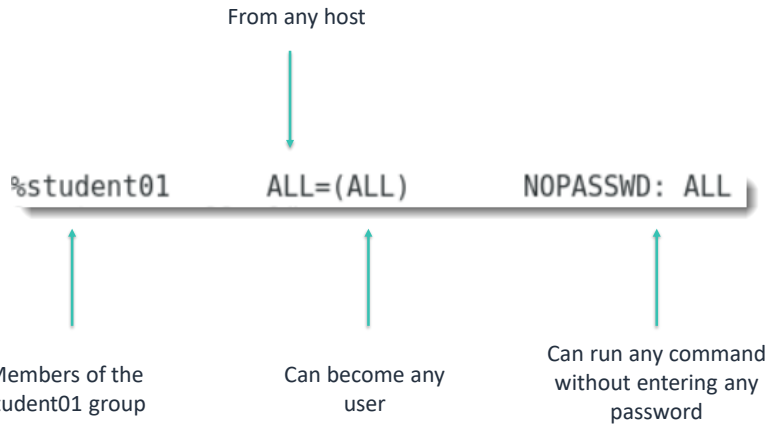
The `/etc/sudoers` file

- Delegate specific commands to specific users by adding them to `/etc/sudoers`
- The syntax is *users hosts=(user:group) commands*
- Examples:
  - Allow members of the users group to shut down the local host:
    - `%users localhost=/usr/sbin/shutdown -r now`
  - Allow members of the devs group all actions from any host without requiring any password:
    - `%devs ALL=(ALL) NOPASSWD: ALL`
- Delegate specific commands to specific users by adding them to the `/etc/sudoers` file

Note: this file must be edited by using the `visudo` command. It must not be edited directly.

The /etc/sudoers file –cont.

Generic format is *# WHO WHERE = (AS WHOM) WHAT*



aws re/start

This diagram depicts the /etc/sudoers file.

Using sudo

The student01 account was delegated the ability to create users by using sudo:

```
[student01@server00 ~]$ sudo useradd user20
[sudo] password for student01:
[student01@server00 ~]$
```

aws re/start

sudo requires the password of the current user whereas su requires the password of the substitute account.

Both of the following commands can be used to add a user:

- `[student01@server00 ~]$ sudo useradd user20`
  - Requires the password of `student01`
  - `student01` must be a sudoers
- `[student01@server00 ~]$ su adminuser`
  `[adminuser@server00 student01]$ useradd user20`
  - `student01` must know the password of the adminuser (could be root or another user with administrative privileges)

sudo is a safer method to run commands because it does not require a password exchange (student01 does not need to know the password of adminuser).

The sudo command

Use `-lU` options to see your delegated `sudo` permissions.

```
[student01@server00 ~]$ sudo -lU student01
Matching Defaults entries for student01 on server00:
    !visiblepw, always_set_home, match_group_by_gid, env_reset,
    env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS",
    env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
    env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES",
    env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY",
    secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User student01 may run the following commands on server00:
    (ALL) NOPASSWD: ALL
[student01@server00 ~]$
```


aws re/start

By using sudo, student01 can run all the commands without any authentication that is needed on the server00 host.

The sudo logs

- The use of sudo permissions is logged at /var/log/messages
- A command that is run with sudo permissions is logged at /var/log/secure

```
[root@server00 ~]# tail /var/log/messages
Feb 28 01:02:16 server00 su: (to root) userA on pts/0
```


aws re/start

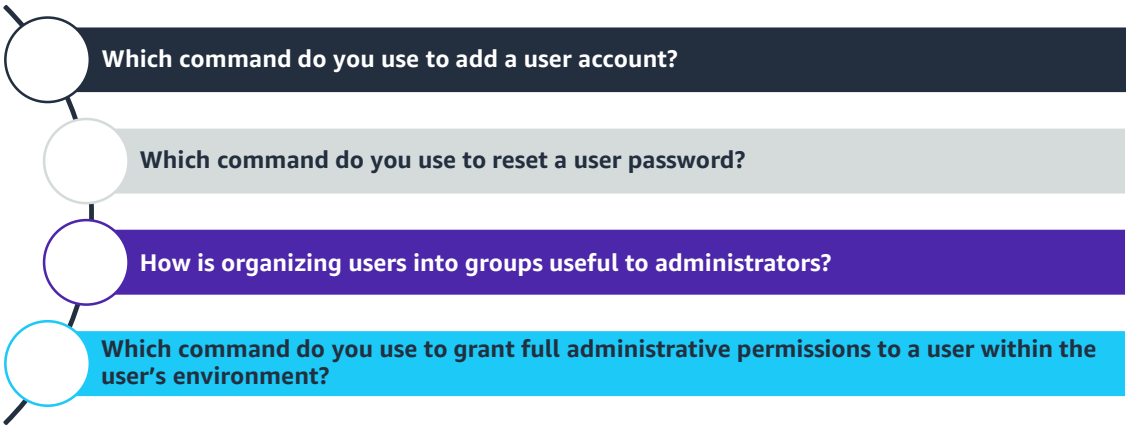
Any time a sudo permission is used, it is logged as shown.

The su command versus the sudo command

- The su command activates full administrative permissions.
  - Used when all administrative permissions are needed
  - Users are prompted for the root password
- The sudo command activates only delegated permissions.
  - Is used to delegate a specific administrative task to a specific standard user
  - Users are prompted for their own password

It is important to understand the differences between the su and sudo commands.

Checkpoint questions



aws re/start

Answers:

1. You use the **useradd** command to add a new user account.
2. You use the **passwd** command to reset a user password.
3. Organizing users into groups allows you to manage their permissions as a single unit.
4. The **su root** command elevates a user to have root permissions in the user's environment.

IAM

In this section, you'll learn about IAM.

AWS Identity and Access Management (IAM)

- IAM is an AWS service that is used to manage users and access to resources
- You can create users, groups, and roles and apply policies to control access to resources
- Access to IAM can be done through:
  - AWS Management Console, a web interface via a browser
  - AWS Command Line Interface (AWS CLI), a command line interface accessible by using a Linux shell or Windows command line
  - AWS software development kits (SDKs) available for many languages, including Java, Python, JavaScript

AWS Identity and Access Management (IAM)

AWS Management Console

AWS Command Line Interface (AWS CLI)

AWS Tools and SDKs

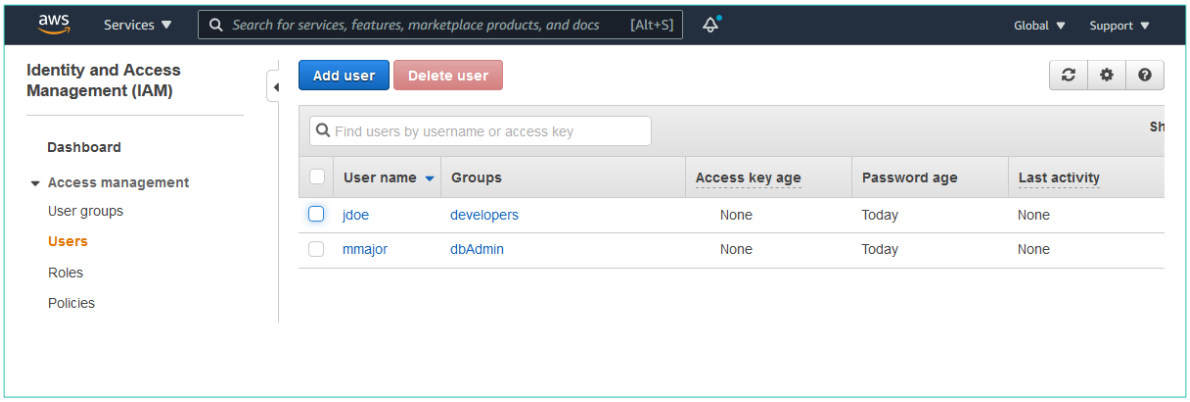
aws re/start

Think of IAM as the tool to centrally manage access. It determines who can launch, configure, manage, and remove resources. It provides control over access permissions for people and for systems or other applications that might make programmatic calls to AWS resources.

A policy is a document that you can attach to a user or a group of users and define the permission to access resources. Examples of such permissions might include administrator access to Amazon Relational Database Service (Amazon RDS) or read-only access to Amazon Simple Storage Service (Amazon S3).

IAM will be covered in more detail later.

A snapshot of the AWS Management Console



aws re/start

The image shows a snapshot of the AWS Management Console.

Key takeaways

- A Linux **user account** represents a user on the system.
- Multiple user accounts can be grouped into a Linux **group** to facilitate the administration of security.
- The **root** user has the permissions to access and modify anything on the system.
- You can use the `su` command to switch to another user to run a command.
- You can use the `sudo` command to run a command with one-time root permissions.
- IAM is an AWS service that is used to manage users and access to resources.

aws re/start

Some key takeaways from this lesson include:

- A Linux user account represents a user on the system.
- Multiple user accounts can be grouped into a Linux group to facilitate the administration of security.
- The root user has the permissions to access and modify anything on the system.
- You can use the `su` command to switch to another user to run a command.
- You can use the `sudo` command to run a command with one-time root permissions.
- IAM is an AWS service that is used to manage users and access to resources.

Thank you

aws re/start

Thank you.