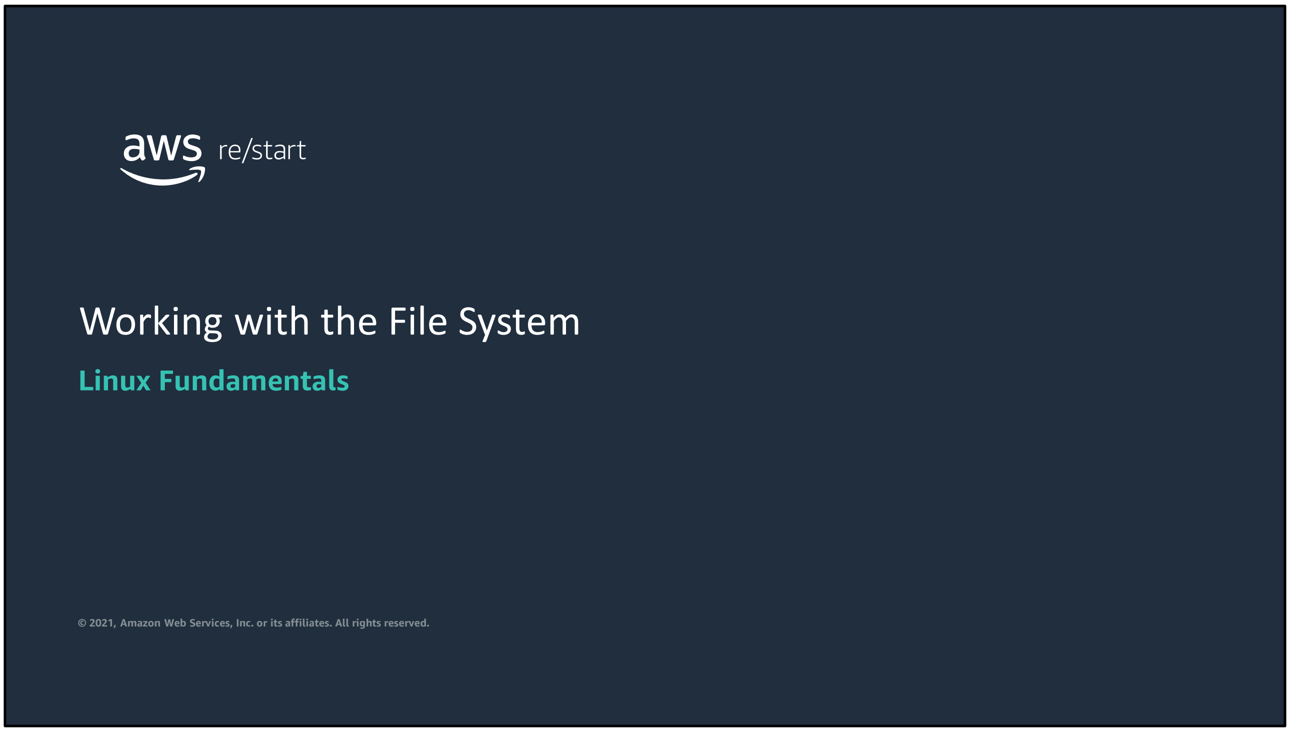
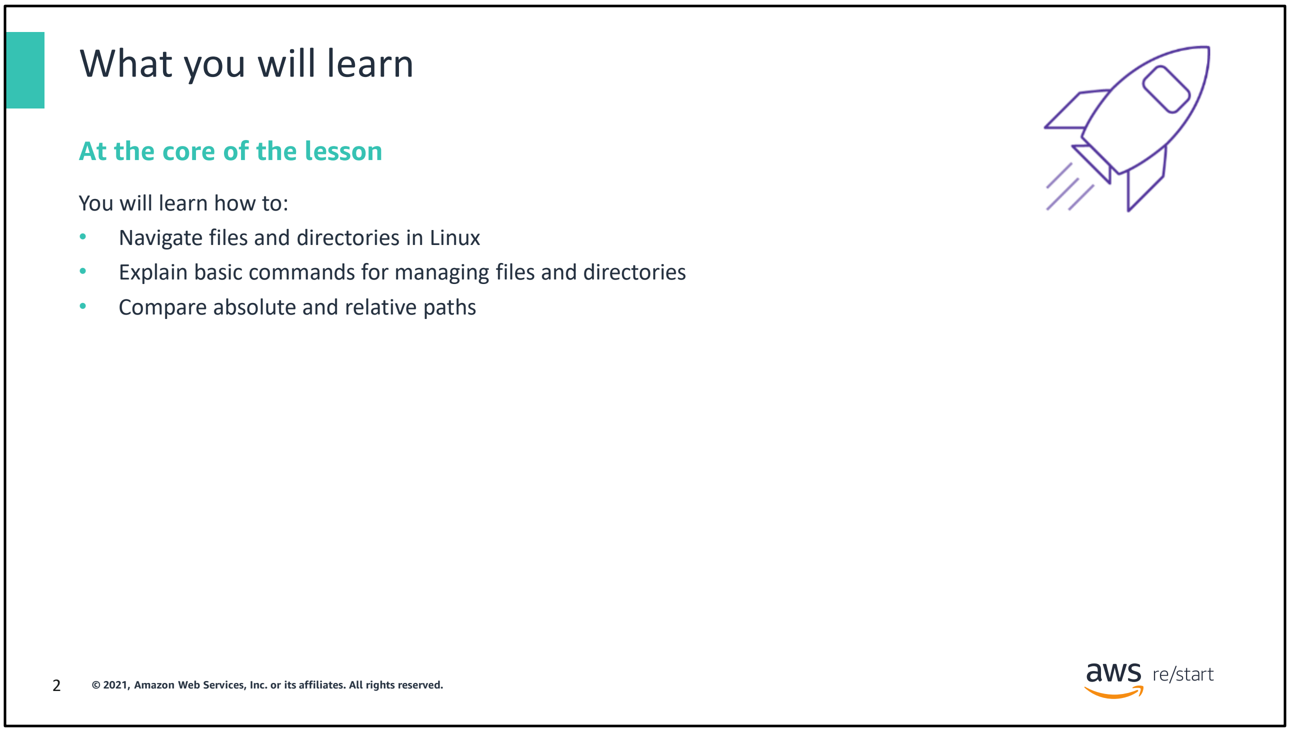


aws re/start

Working with the File System

Linux Fundamentals

Welcome to Working with the File System.



What you will learn

At the core of the lesson

You will learn how to:

- Navigate files and directories in Linux
- Explain basic commands for managing files and directories
- Compare absolute and relative paths

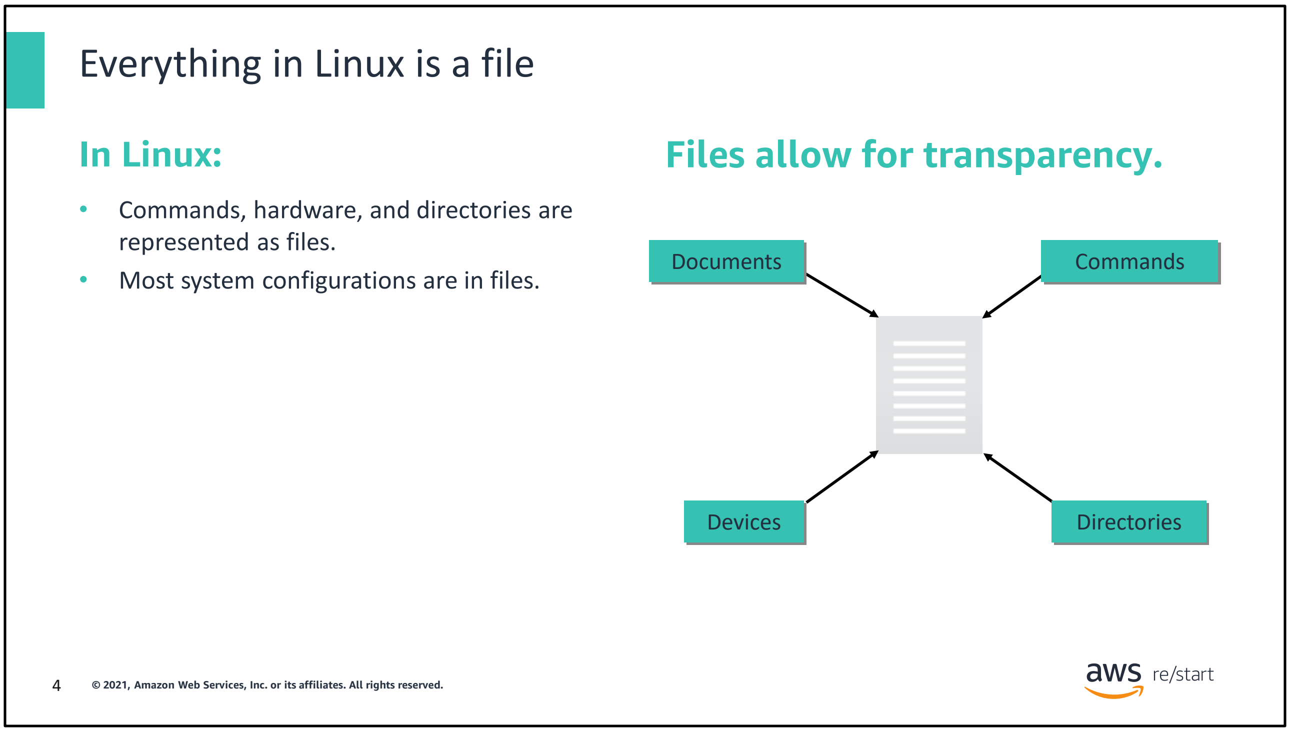
aws re/start

In this lesson, you will learn how to:

- Navigate files and directories in Linux
- Explain basic commands for managing files and directories
- Compare absolute and relative paths

Navigating files and directories

Introducing the Linux file system.

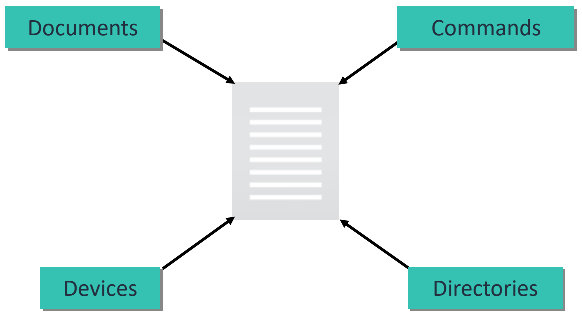


Everything in Linux is a file

In Linux:

- Commands, hardware, and directories are represented as files.
- Most system configurations are in files.

Files allow for transparency.

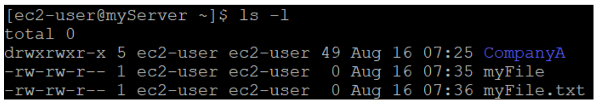


Files allow for transparency. Drives, processes, and other elements are all represented as files. They can be browsed and accessed for information (for example, `ls /proc` gives you access to processes).

Files allow for interoperability. The same tools can be used for different types of files and can be combined (for example, `ls -l  | grep .txt`).

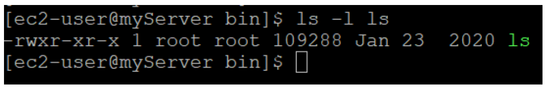
Files

A **directory**



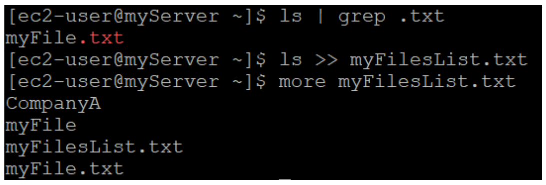
```
[ec2-user@myServer ~]$ ls -l
total 0
drwxrwxr-x 5 ec2-user ec2-user 49 Aug 16 07:25 CompanyA
-rw-rw-r-- 1 ec2-user ec2-user  0 Aug 16 07:35 myFile
-rw-rw-r-- 1 ec2-user ec2-user  0 Aug 16 07:36 myFile.txt
```

`ls` command with the option `-l` can list the `ls` command configuration file because it is also a file.



```
[ec2-user@myServer bin]$ ls -l ls
-rwxr-xr-x 1 root root 109288 Jan 23  2020 ls
[ec2-user@myServer bin]$
```

The `ls` command can be used with other commands to create a search condition for .txt documents.



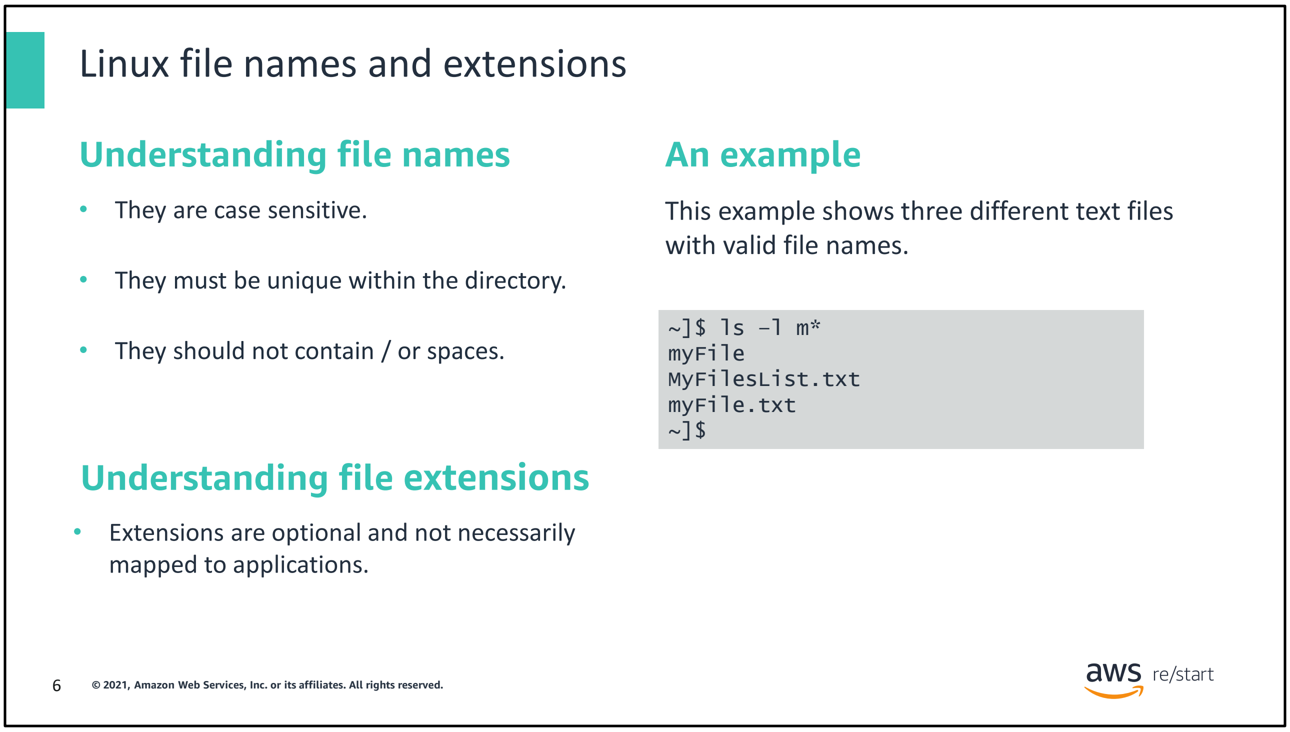
```
[ec2-user@myServer ~]$ ls | grep .txt
myFile.txt
[ec2-user@myServer ~]$ ls >> myFilesList.txt
[ec2-user@myServer ~]$ more myFilesList.txt
CompanyA
myFile
myFilesList.txt
myFile.txt
```

aws re/start

In the first screenshot, both the directory and the text file are considered files. The directory is a special kind of file, hence the d and the blue color.

In the second screenshot, you see that you can list the `ls` command because it is also a file.

The third screenshot shows how you can operate between commands.



Linux file names and extensions

Understanding file names

- They are case sensitive.
- They must be unique within the directory.
- They should not contain / or spaces.

Understanding file extensions

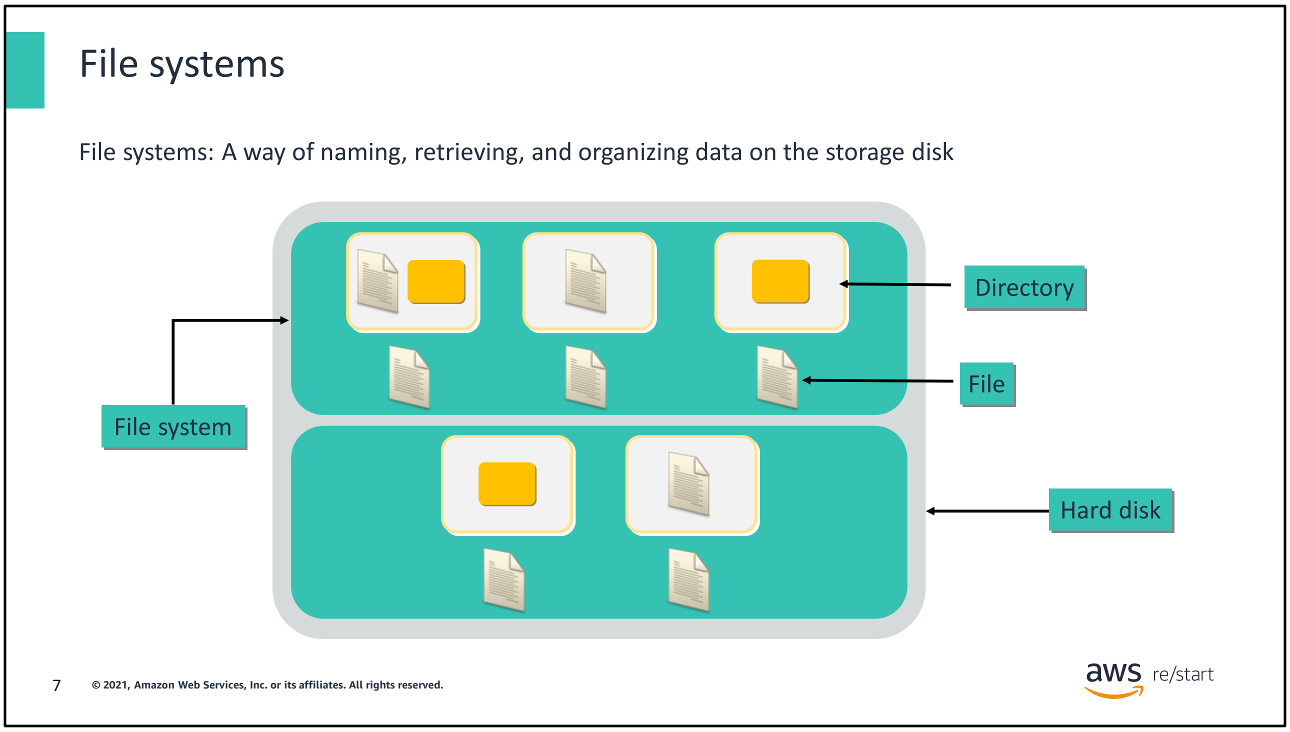
- Extensions are optional and not necessarily mapped to applications.

An example

This example shows three different text files with valid file names.

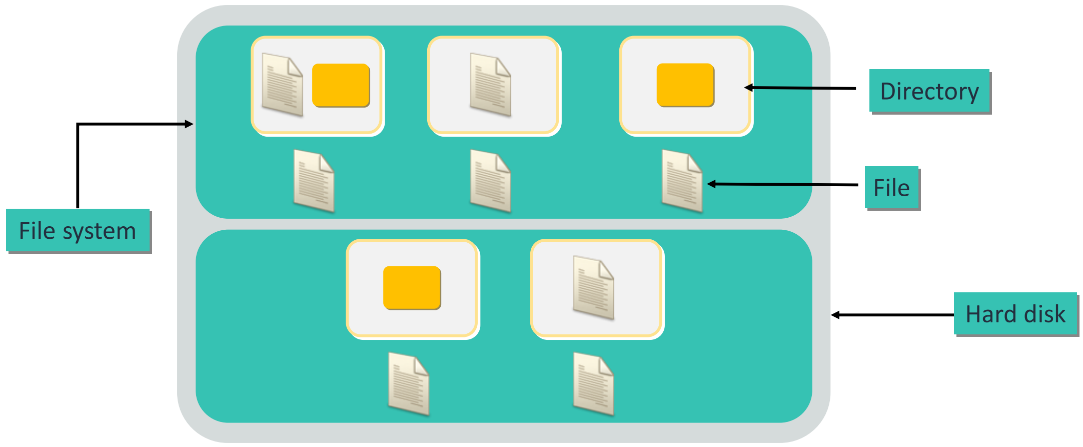
```
~]$ ls -l m*
myFile
MyFilesList.txt
myFile.txt
~]$
```

You are strongly advised to have consistent extensions. For instance, a .jpg image could be named image.txt even though this extension would not make sense. A user might think that this file is a description file and try to open it with a text editor instead of an image viewer. A better option is to name it image.jpeg or image.jpg.

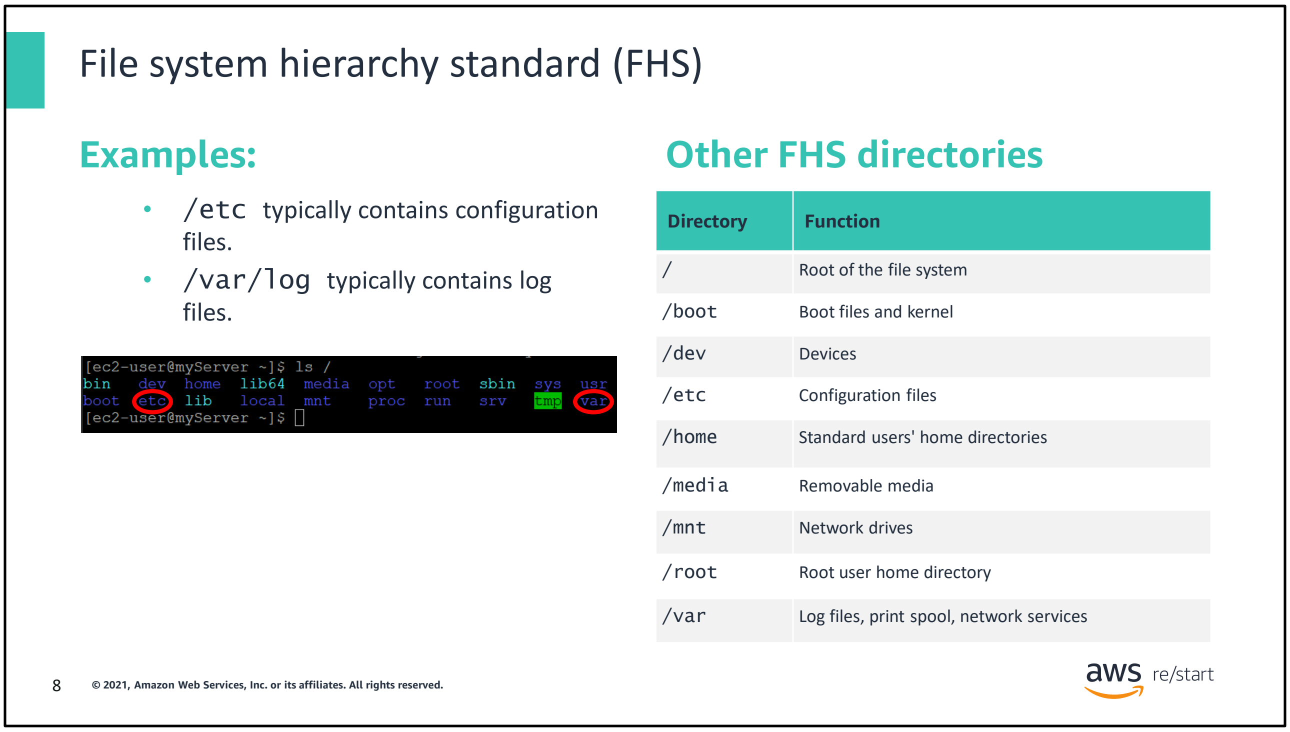


File systems

File systems: A way of naming, retrieving, and organizing data on the storage disk



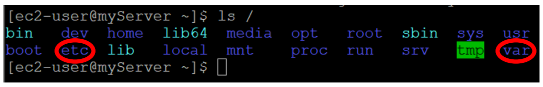
The file system organizes how files are stored on the hard drive. A file is located inside a directory.



File system hierarchy standard (FHS)

Examples:

- `/etc` typically contains configuration files.
- `/var/log` typically contains log files.



```
[ec2-user@myServer ~]$ ls /
bin   dev  home  lib64  media  opt   root  sbin  sys  usr
boot  etc  lib   local  mnt    proc  run   srv   tmp  var
[ec2-user@myServer ~]$ 
```

Other FHS directories

Directory	Function
/	Root of the file system
/boot	Boot files and kernel
/dev	Devices
/etc	Configuration files
/home	Standard users' home directories
/media	Removable media
/mnt	Network drives
/root	Root user home directory
/var	Log files, print spool, network services

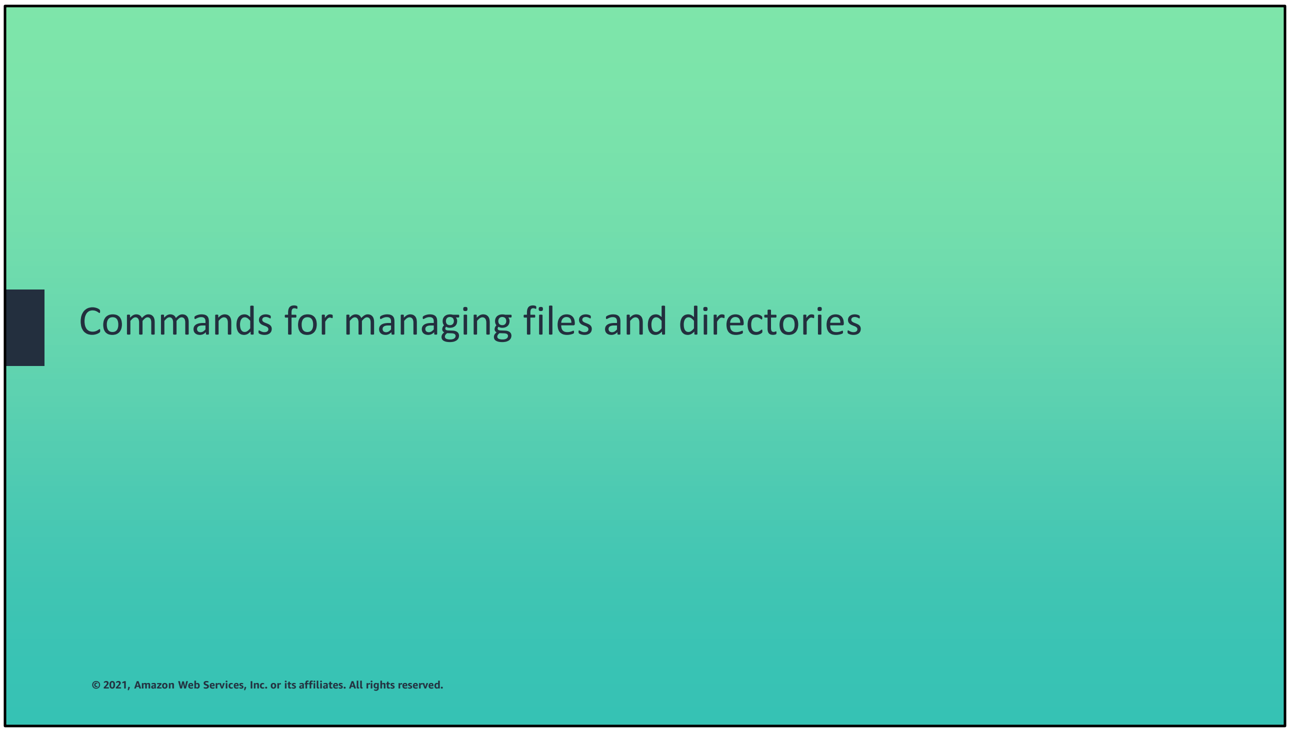
Most Linux distributions use this standard, but some distributions might differ slightly or intentionally use a different file system.

There are some commonalities between some of these distributions. For example, there are standard locations and functions of directories across Linux distribution.

Most Linux distributions:
- Allow software to be compatible with various distributions
- Allow administrators to predict where certain types of files will be found

Most distributions follow the file system hierarchy standard (FHS).

The FHS has many other directories. The table shows a list of the other available directories.



Commands for managing files and directories

The following commands are important. They are used to manage files and directories.

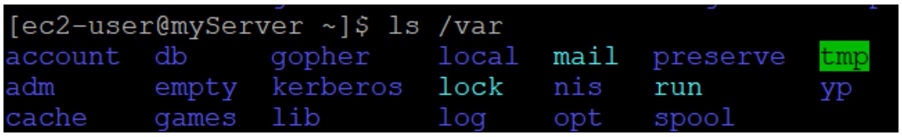
Understanding command syntax with the `ls` command

ls command

- The `ls` command displays a list of files in a directory.

What the command does

- Different colors represent different types of files.
- `ls` command lists the content of the current directory.
- `ls dir` command lists the content of the `dir` directory.



```
[ec2-user@myServer ~]$ ls /var
account  db     gopher    local  mail  preserve  tmp
adm      empty  kerberos  lock   nis   run       yp
cache    games  lib       log    opt   spool
```

ls /var command output

Colors are not defined according to a standard. They depend on the configuration of the shell that you are using.

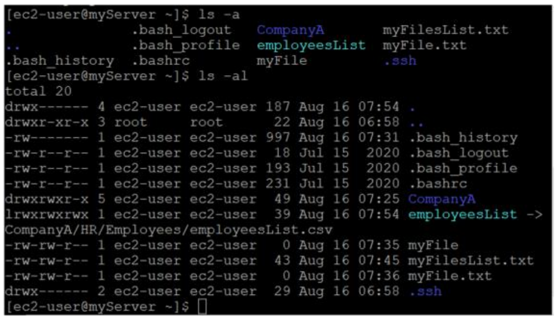
You can list several multiple directories, for example, `ls directory1 directory2`.

`ls` command options and examples

Useful options

Option	Description
`-l`	Long format (shows permissions)
`-h`	File sizes reported in a human-friendly format
`-a`	Shows all files, including hidden files
`-R`	Lists subdirectories
`--sort=extension` or `-X`	Sorts alphabetically by file extension
`--sort=size` or `-S`	Sorts by file size
`--sort=time` or `-t`	Sorts by modification time
`--sort=version` or `-v`	Sorts by version number

Examples of the ls command



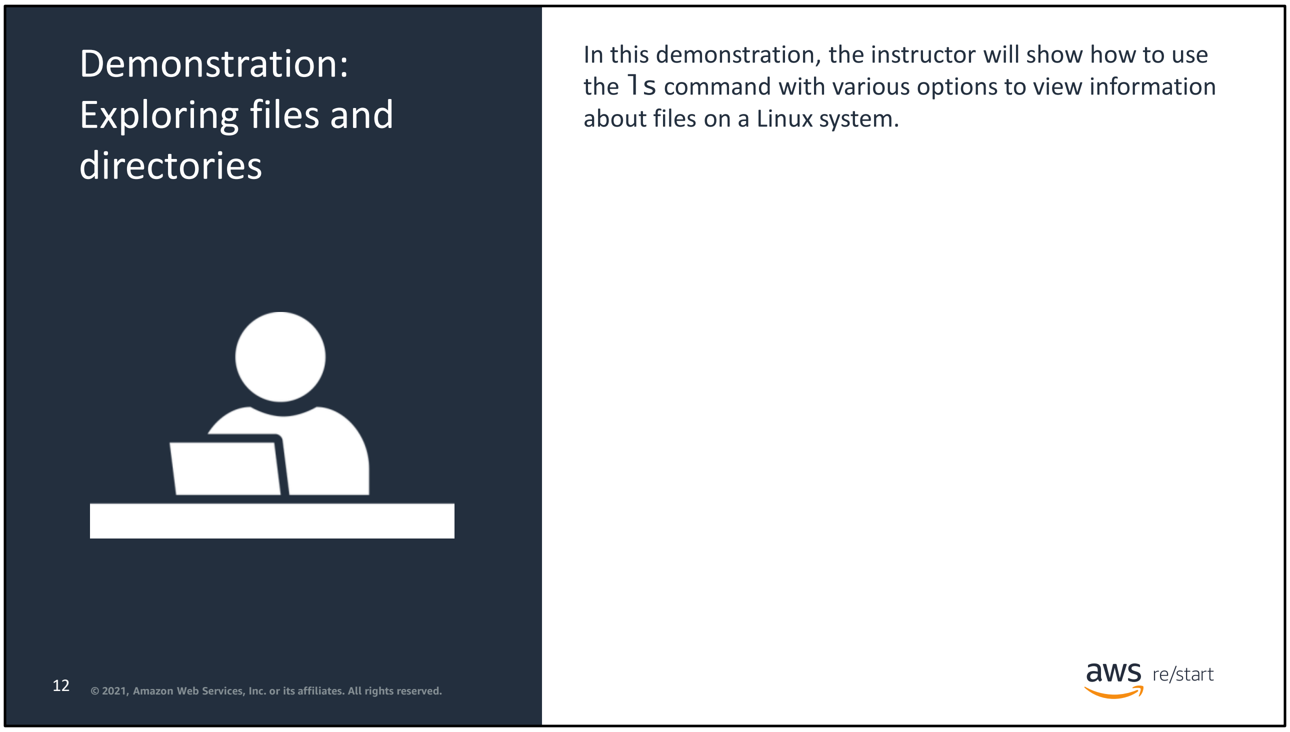
```
[ec2-user@myServer ~]$ ls -a
.                 .bash_logout   CompanyA        myFilesList.txt
..                .bash_profile  employeesList   myFile.txt
.bash_history  .bashrc        myFile          .ssh
[ec2-user@myServer ~]$ ls -al
total 20
drwx------ 4 ec2-user ec2-user 187 Aug 16 07:54 .
drwxr-xr-x 3 root     root      22 Aug 16 06:58 ..
-rw------- 1 ec2-user ec2-user 997 Aug 16 07:31 .bash_history
-rw-r--r-- 1 ec2-user ec2-user  18 Jul 15  2020 .bash_logout
-rw-r--r-- 1 ec2-user ec2-user 193 Jul 15  2020 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 231 Jul 15  2020 .bashrc
drwxrwxr-x 5 ec2-user ec2-user  49 Aug 16 07:25 CompanyA
lrwxrwxrwx 1 ec2-user ec2-user  39 Aug 16 07:54 employeesList ->
CompanyA/HR/Employees/employeesList.csv
-rw-rw-r-- 1 ec2-user ec2-user   0 Aug 16 07:35 myFile
-rw-rw-r-- 1 ec2-user ec2-user  43 Aug 16 07:45 myFilesList.txt
-rw-rw-r-- 1 ec2-user ec2-user   0 Aug 16 07:36 myFile.txt
drwx------ 2 ec2-user ec2-user  29 Aug 16 06:58 .ssh
[ec2-user@myServer ~]$ 
```

aws re/start

By default, the ls command uses the natural sort order. To get results in the reverse order, add the **–r** option.

You can combine options: `ls –al` displays hidden files and file details.

- `ls –l` lists the contents of the current directory with details. It does not display the hidden files.
- `ls –a` displays the hidden files.
- `ls –al` displays the hidden files and the file's details (not all of the list is displayed on the screenshot because it would take too much space).



Demonstration: Exploring files and directories

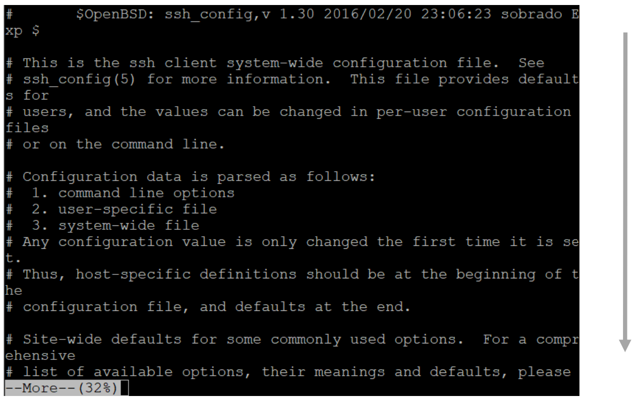
In this demonstration, the instructor will show how to use the `ls` command with various options to view information about files on a Linux system.

aws re/start

Follow along as the instructor demonstrates the use of the `ls` command.

more command

- Used to view file contents that don't fit on one screen.
- Loads entire contents of files before displaying results
- Can only scroll down
- Can be used in conjunction with other commands:
  `cat file.txt | more`



```
#       $OpenBSD: ssh_config,v 1.30 2016/02/20 23:06:23 sobrado E
xp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides default
s for
# users, and the values can be changed in per-user configuration
files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is se
t.
# Thus, host-specific definitions should be at the beginning of t
he
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a compr
ehensive
# list of available options, their meanings and defaults, please
--More--(32%)
```

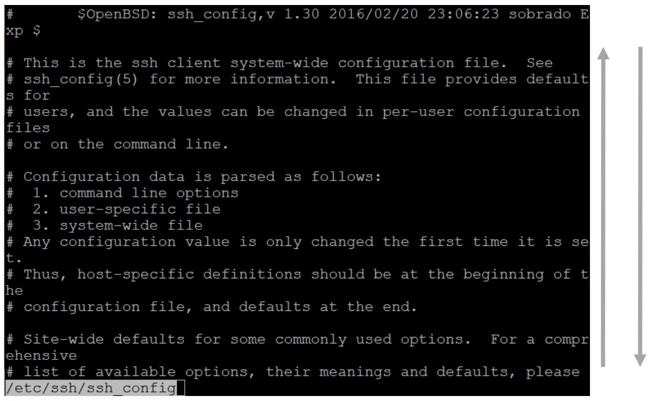
Usage:

```
more [-options] [-num] [+/pattern] [+linenum] [file_name]
```

- Options:
  - `-d`: Displays information about how to navigate at the bottom of the screen
  - `-f`: Prevents line wrap
  - `-p`: Clears the screen before displaying the content
  - `-s`: Squeezes multiple blank lines into one line
- `num`: Number of lines to display
- `/pattern`: String to find in the file
- `linenum`: The line number where the content starts to display
- `file_name`: Name of the file to display the content of

`less` command

- Displays file contents that don't fit on one screen
- Can scroll up and down through content
- Loads faster than `more` because `less` doesn't load every page before it displays results
- Used mostly for large files



```
#       $OpenBSD: ssh_config,v 1.30 2016/02/20 23:06:23 sobrado E
xp $

# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides default
s for
# users, and the values can be changed in per-user configuration
files
# or on the command line.

# Configuration data is parsed as follows:
#  1. command line options
#  2. user-specific file
#  3. system-wide file
# Any configuration value is only changed the first time it is se
t.
# Thus, host-specific definitions should be at the beginning of t
he
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options.  For a compr
ehensive
# list of available options, their meanings and defaults, please
/etc/ssh/ssh_config
```

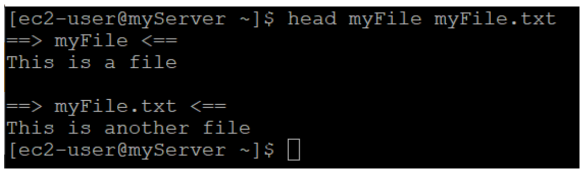
Usage:

`less [OPTIONS] filename`

- Use Q to quit.
- Options:
  - –N: Shows line numbers
  - –X: Displays the content after the last command and does not clear the screen when exiting
  - +F: Watches for file content changes

head command

- Displays the first 10 lines of a file by default
- Can display multiple files

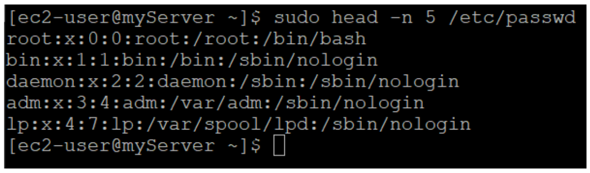


```
[ec2-user@myServer ~]$ head myFile myFile.txt
==> myFile <==
This is a file

==> myFile.txt <==
This is another file
[ec2-user@myServer ~]$
```

Head myFile my File.txt

When the head command is used in conjunction with the -n option, you can specify the number of lines to display.



```
[ec2-user@myServer ~]$ sudo head -n 5 /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
[ec2-user@myServer ~]$
```

Sudo head -n 5 /etc.passwd

aws re/start

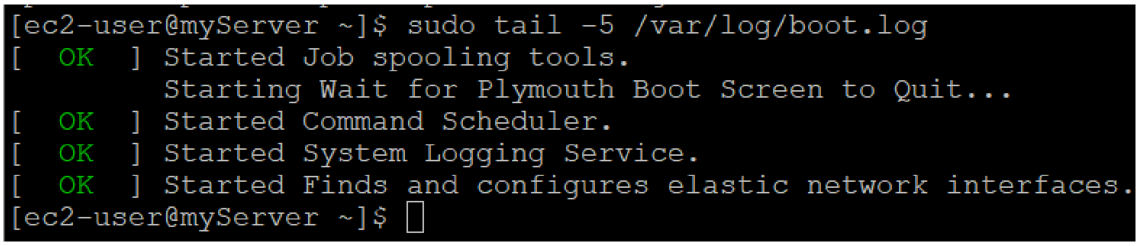
Usage:

`head [OPTIONS] filename(s)`

- Options:
  - `-n <number>`: First n lines to display
  - `-c <number>`: First c bytes to display

`tail` command

- Displays the last 10 lines of a file by default
- Use the `tail` command with the `-n` option to specify the number of lines to display.



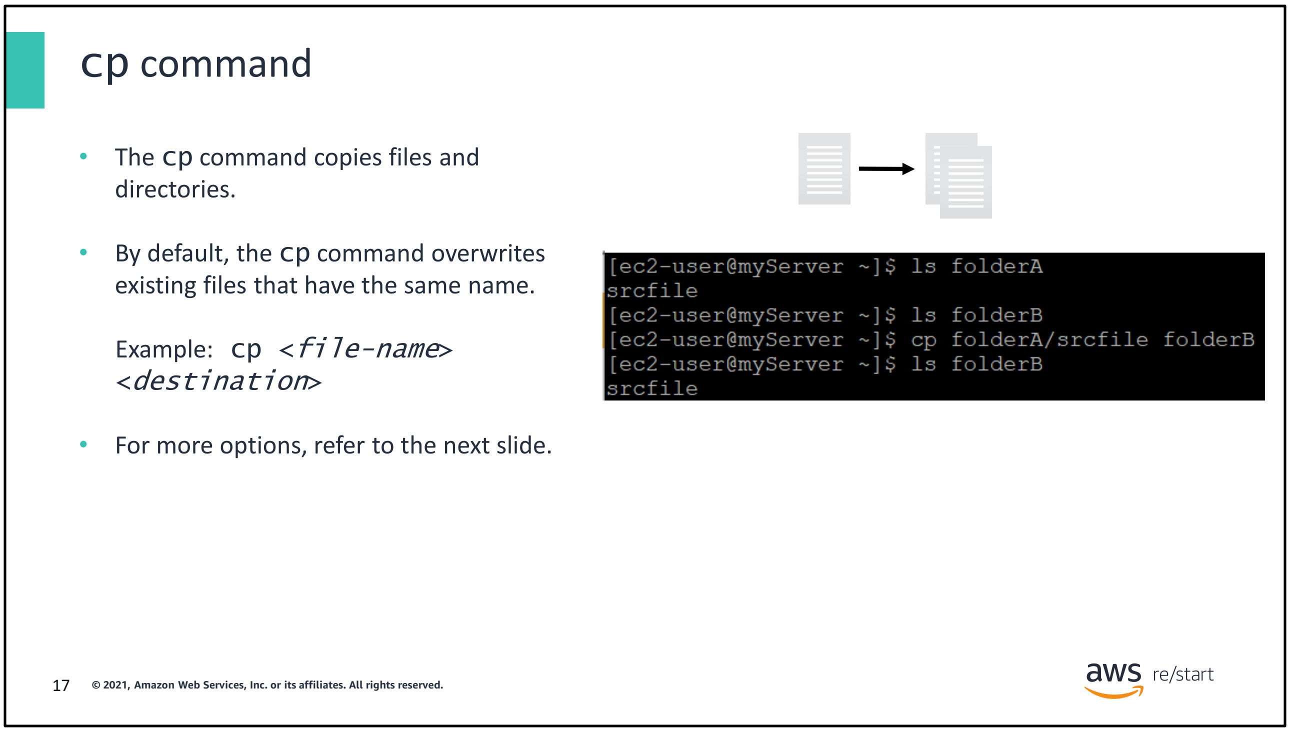
```
[ec2-user@myServer ~]$ sudo tail -5 /var/log/boot.log
[  OK  ] Started Job spooling tools.
         Starting Wait for Plymouth Boot Screen to Quit...
[  OK  ] Started Command Scheduler.
[  OK  ] Started System Logging Service.
[  OK  ] Started Finds and configures elastic network interfaces.
[ec2-user@myServer ~]$ 
```

Usage:

`tail [OPTIONS] filename(s)`

- Options:
  - `-n <number>`: Last `n` lines to display
  - `-c <number>`: Last `c` bytes to display
  - `-f`: Monitor for file changes

The `tail -f` command is useful for log files that are regularly updated and where the most recent entries are at the bottom of the file.



cp command

- The cp command copies files and directories.
- By default, the cp command overwrites existing files that have the same name.

  Example: `cp <file-name> <destination>`
- For more options, refer to the next slide.

```
[ec2-user@myServer ~]$ ls folderA
srcfile
[ec2-user@myServer ~]$ ls folderB
[ec2-user@myServer ~]$ cp folderA/srcfile folderB
[ec2-user@myServer ~]$ ls folderB
srcfile
```

17 © 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws re/start

Usage:

`cp  folderA/srcfile folderB/destfile`

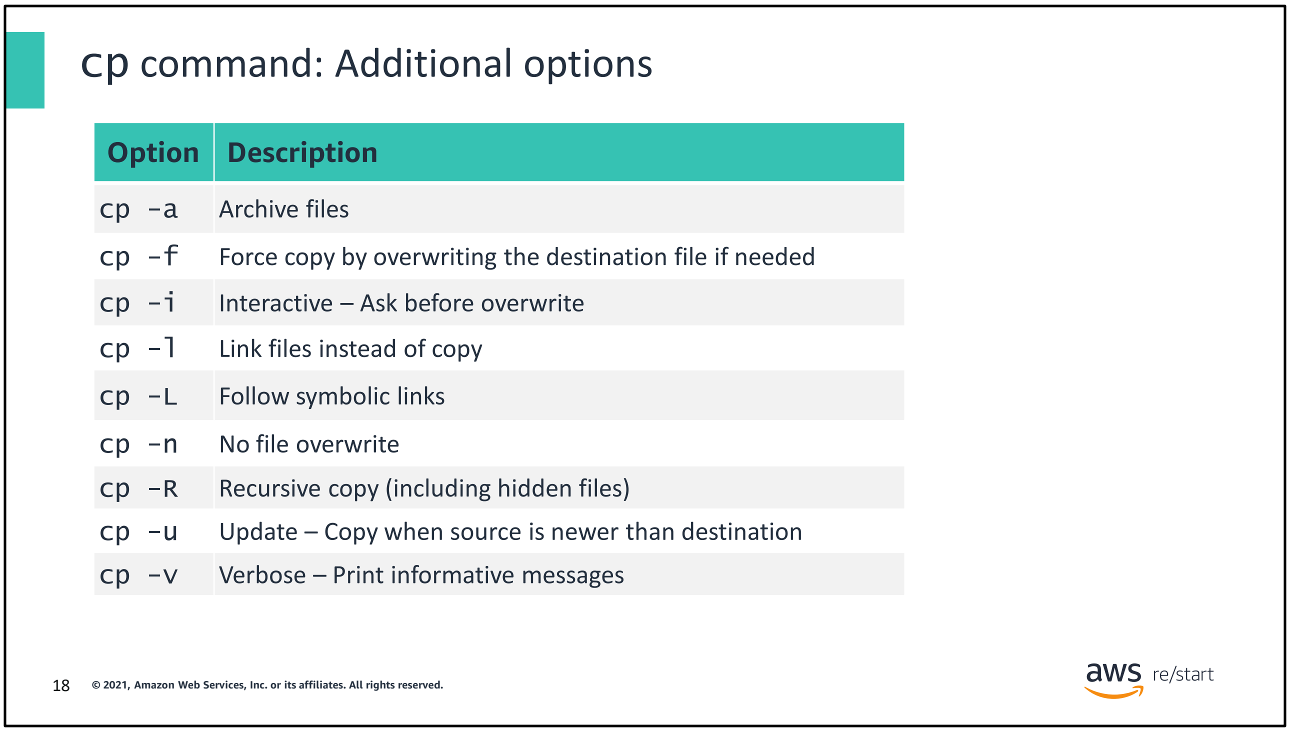
- Copies the srcfile that is located in folderA to folderB and names it destfile

`cp  folderA/srcfile folderB/`

- Copies the srcfile that is located in folderA to folderB (and both files have the same name)

`cp  folderA/srcfile folderB/ folderC/destfile`

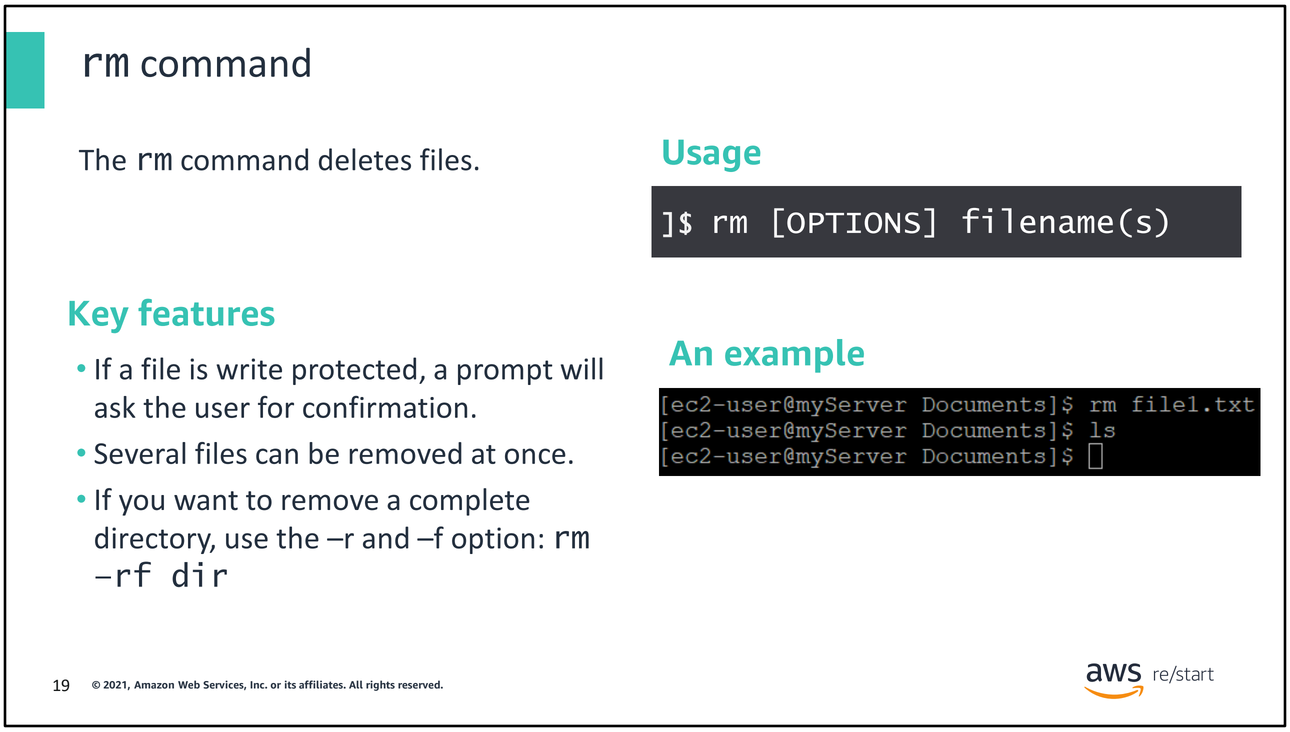
- Copies the srcfile that is located in folderA to folderB and to folderC with the name destfile



cp command: Additional options

Option	Description
`cp -a`	Archive files
`cp -f`	Force copy by overwriting the destination file if needed
`cp -i`	Interactive – Ask before overwrite
`cp -l`	Link files instead of copy
`cp -L`	Follow symbolic links
`cp -n`	No file overwrite
`cp -R`	Recursive copy (including hidden files)
`cp -u`	Update – Copy when source is newer than destination
`cp -v`	Verbose – Print informative messages

This slide shows additional options for the cp command that you can use.



rm command

The `rm` command deletes files.

Usage

```
]$ rm [OPTIONS] filename(s)
```

Key features

- If a file is write protected, a prompt will ask the user for confirmation.
- Several files can be removed at once.
- If you want to remove a complete directory, use the –r and –f option: `rm -rf dir`

An example

```
[ec2-user@myServer Documents]$ rm file1.txt
[ec2-user@myServer Documents]$ ls
[ec2-user@myServer Documents]$
```

Usage:

`rm [OPTIONS] filename(s)`

- Options:
  - `-d`: Removes a directory; the directory must be empty: `rm -d dir`
  - `-r`: Allows you to remove a non-empty directory: `rm -r dir`
  - `-f`: Never prompt user (useful when deleting a directory with many files)
  - `-i`: Prompts the user for confirmation for each file
  - `-v`: Display the names of deleted files

- If a file is write protected, a prompt will ask the user for confirmation.
- Several files can be removed at once.
- If you want to remove a complete directory, use the –r and –f option: `rm -rf dir`
- You can use a regular expression: `rm *.png` removes all files that end with .png.

`mkdir` command

The `mkdir` command creates new directories.

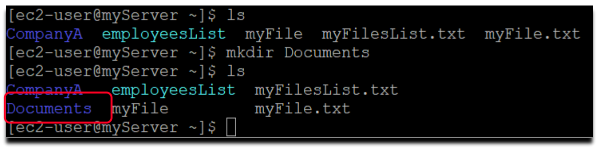
Options

- `-m <mask>`: Sets a permission to the directory
- `-p`: Creates a parent directory

Usage

```
]$ mkdir [OPTIONS] filename(s)
```

An example



```
[ec2-user@myServer ~]$ ls
CompanyA  employeesList  myFile  myFilesList.txt  myFile.txt
[ec2-user@myServer ~]$ mkdir Documents
[ec2-user@myServer ~]$ ls
CompanyA   employeesList  myFilesList.txt
Documents  myFile         myFile.txt
[ec2-user@myServer ~]$
```

aws re/start

Usage:

`mkdir [OPTIONS] filename(s)`

Options:
- `-m <mask>`: Sets a permission to the directory
- `-p`: Creates a parent directory

You can create several directories with one command: `mkdir dir1 dir2 dir3`.

`mkdir -m 700 dir1` creates the `dir1` directory with the mask 700 for permissions.

`mkdir -p /home/user/dir1/dir2`: If `dir1` does not exist, the creation will fail without the `-p` option.

mv command

The mv command moves a file from one directory to another.

The mv command renames a file if the source and destination are the same

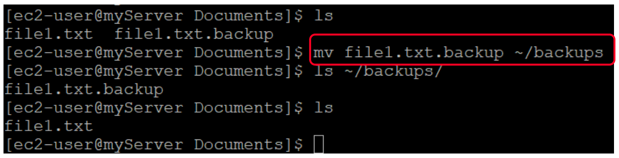
Usage

```
]$ mv [OPTIONS] destination
```

Note:

By default, the mv command overwrites existing files that have the same name.

An example



```
[ec2-user@myServer Documents]$ ls
file1.txt  file1.txt.backup
[ec2-user@myServer Documents]$ mv file1.txt.backup ~/backups
[ec2-user@myServer Documents]$ ls ~/backups/
file1.txt.backup
[ec2-user@myServer Documents]$ ls
file1.txt
[ec2-user@myServer Documents]$ 
```

aws re/start

Usage:

`mv [OPTIONS] source destination`

Options:
- `-i`: Prompts before overwritting a file
- `-f`: Avoids being prompted
- `-n`: Does not overwrite existing files
- `-v`: Verbose option, prints the name of files that are moved or renamed
- `mv file1 dir1`: Moves file1 to dir1
- `mv dir1 dir2`: Moves dir1 to dir2
- `mv file1 file2 dir1 dir2`: Moves file1, file2, and dir1 to dir2; there can only be one target directory here, dir2
- `mv file1 dir1/file2`: Moves file1 to dir1 and renames it file2
- `mv file1 file2`: Renames file1 as file2

You can use a regular expression to move files of the same type:

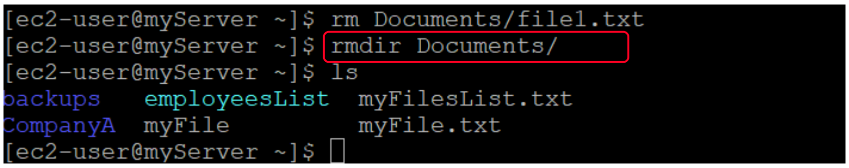
`mv *.png dir1` moves all files with extension .png into dir1.

rmdir command

The rmdir command deletes existing empty directories: rmdir <DirectoryName>

If a directory isn't empty, use rm -r <DirectoryName>.

This command removes a directory and all of its contents.

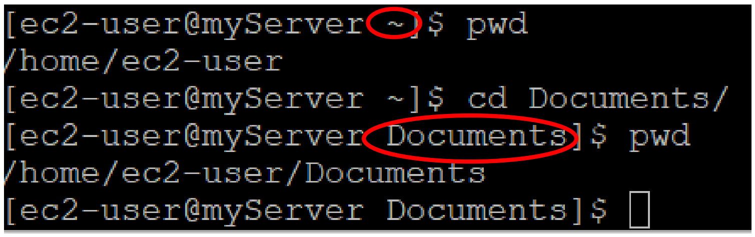


```
[ec2-user@myServer ~]$ rm Documents/file1.txt
[ec2-user@myServer ~]$ rmdir Documents/
[ec2-user@myServer ~]$ ls
backups    employeesList  myFilesList.txt
CompanyA   myFile         myFile.txt
[ec2-user@myServer ~]$
```

rmdir is equivalent to rm –d.

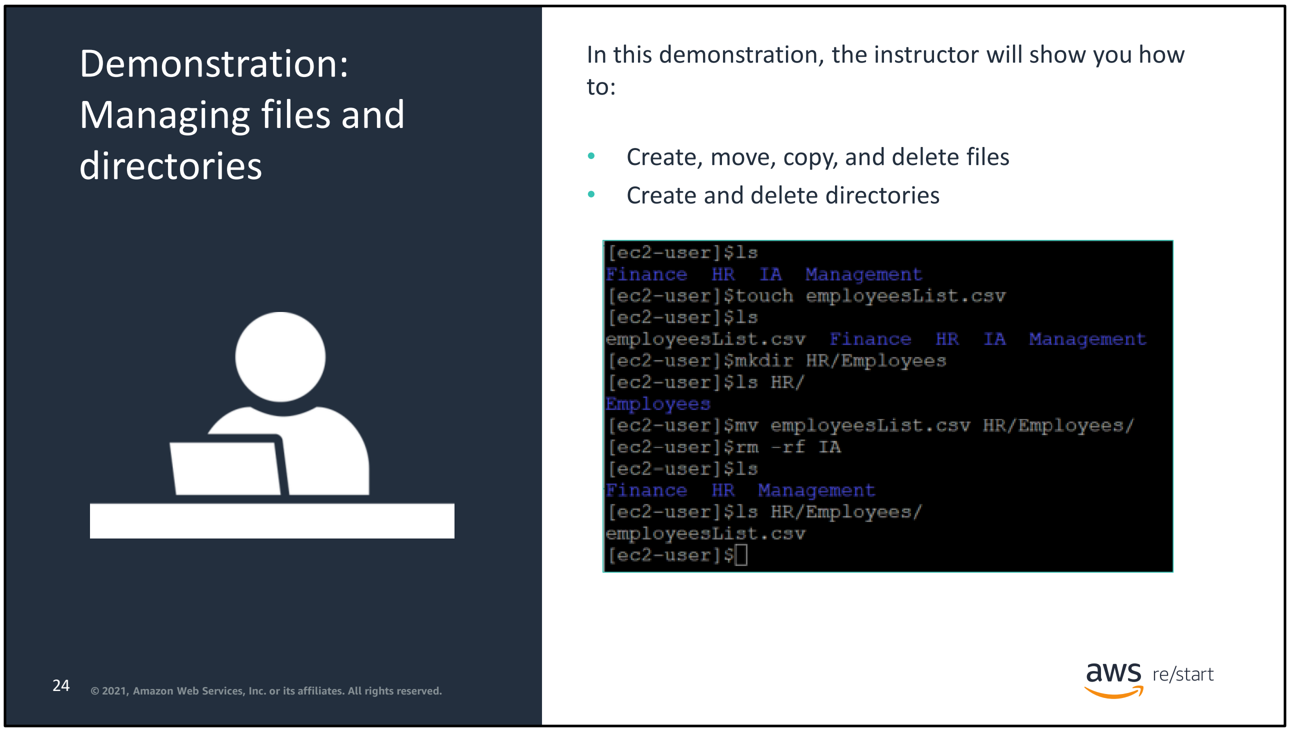
pwd command

- Output of the pwd command: Absolute path to your current location in the file system
- Essential for navigation: You must know where you are in the file system to move to other directories.



```
[ec2-user@myServer ~]$ pwd
/home/ec2-user
[ec2-user@myServer ~]$ cd Documents/
[ec2-user@myServer Documents]$ pwd
/home/ec2-user/Documents
[ec2-user@myServer Documents]$
```

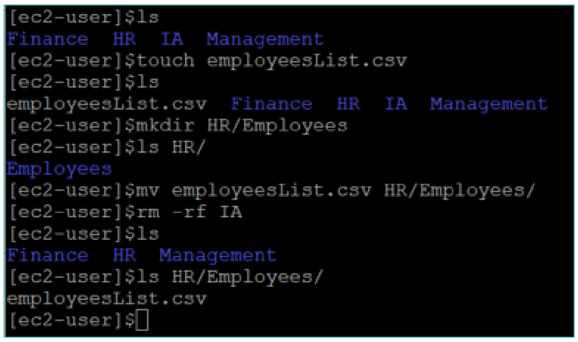
Use the pwd command to know where you are in the file directory structure.



Demonstration: Managing files and directories

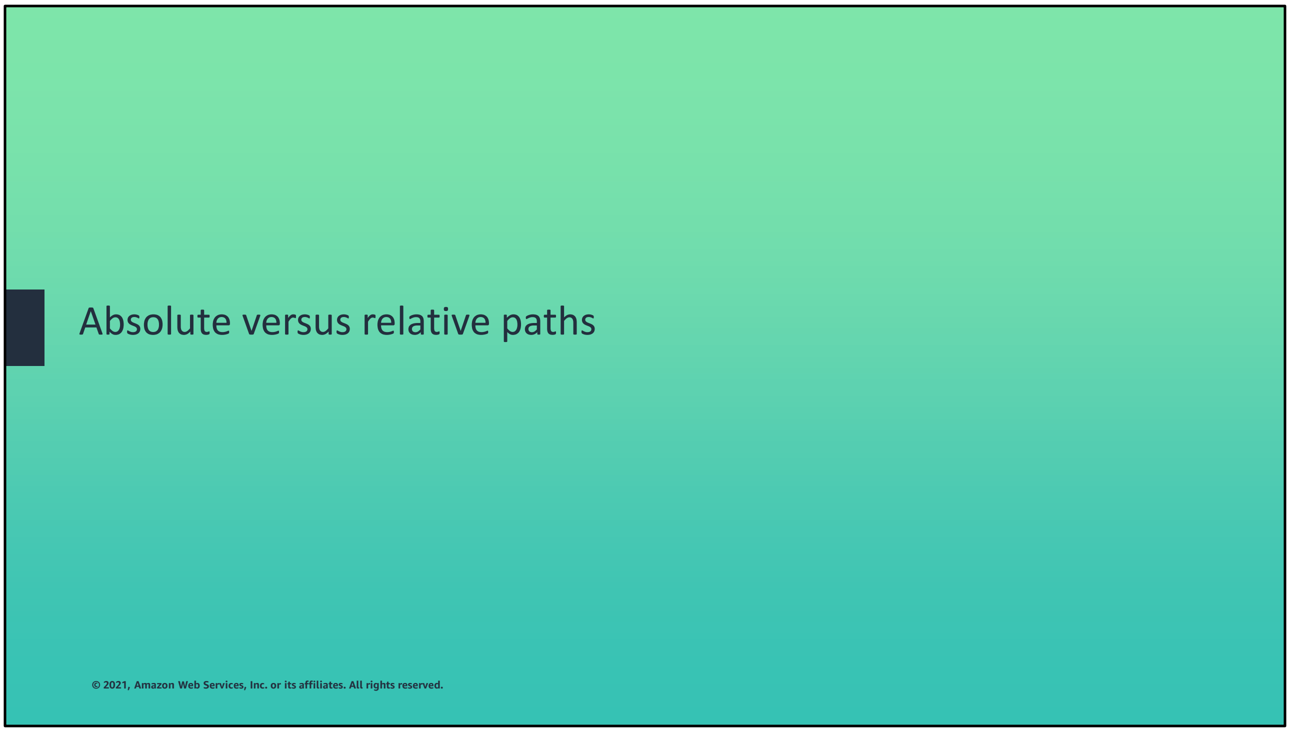
In this demonstration, the instructor will show you how to:

- Create, move, copy, and delete files
- Create and delete directories



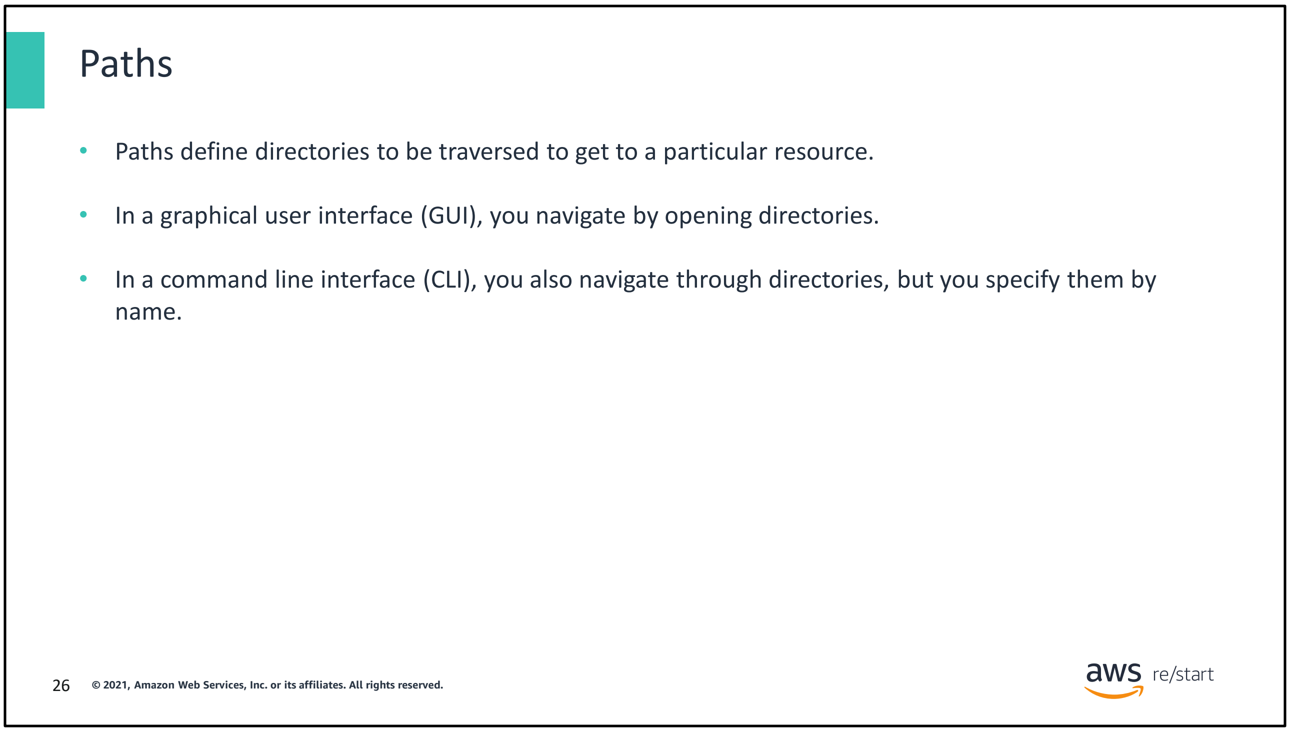
```
[ec2-user]$ls
Finance  HR  IA  Management
[ec2-user]$touch employeesList.csv
[ec2-user]$ls
employeesList.csv  Finance  HR  IA  Management
[ec2-user]$mkdir HR/Employees
[ec2-user]$ls HR/
Employees
[ec2-user]$mv employeesList.csv HR/Employees/
[ec2-user]$rm -rf IA
[ec2-user]$ls
Finance  HR  Management
[ec2-user]$ls HR/Employees/
employeesList.csv
[ec2-user]$
```

Follow along as the instructor creates, moves, copies, and deletes files and creates and deletes directories.



Absolute versus relative paths

You must know the difference between absolute and relative paths.

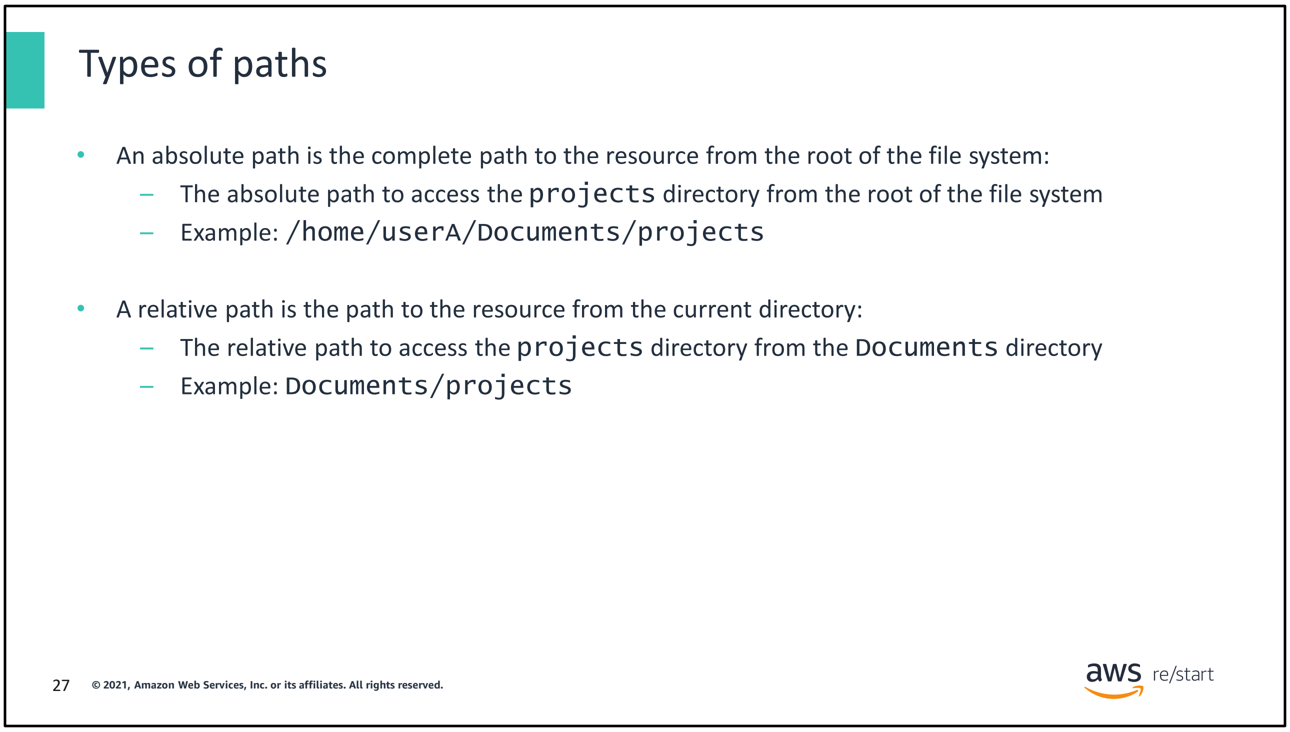


Paths

- Paths define directories to be traversed to get to a particular resource.
- In a graphical user interface (GUI), you navigate by opening directories.
- In a command line interface (CLI), you also navigate through directories, but you specify them by name.

aws re/start

You must know how to navigate directories by both a GUI and a CLI.


Types of paths

- An absolute path is the complete path to the resource from the root of the file system:
  - The absolute path to access the `projects` directory from the root of the file system
  - Example: `/home/userA/Documents/projects`
- A relative path is the path to the resource from the current directory:
  - The relative path to access the `projects` directory from the `Documents` directory
  - Example: `Documents/projects`

Suppose the command pwd tells you that you are in the folder `/home/ec2-user`.

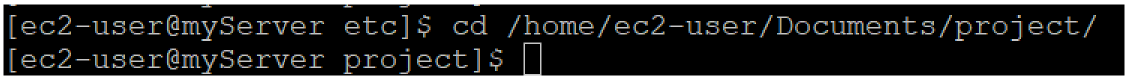
`cd /home/userA/Documents/projects` will navigate to the `/home/userA/Documents/projects` folder.

`cd Documents/projects` will navigate to the `/home/ec2-user/Documents/projects` folder (currentfolder/Document/projects, where current folder is /home/ec2-user).

cd command

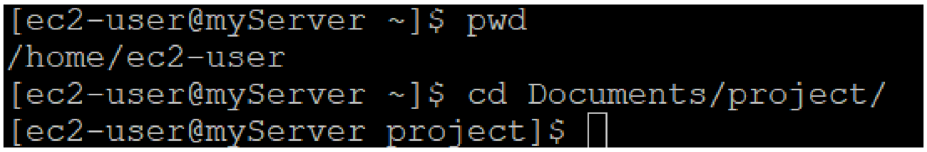
The change directory or cd command is used to move from one directory to another.

- Using the cd command with the absolute path:



```
[ec2-user@myServer etc]$ cd /home/ec2-user/Documents/project/
[ec2-user@myServer project]$ 
```

- Using the cd command with the relative path:



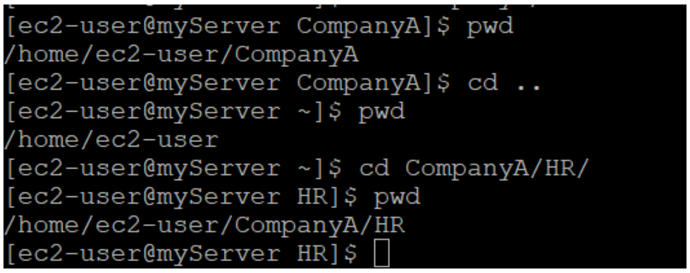
```
[ec2-user@myServer ~]$ pwd
/home/ec2-user
[ec2-user@myServer ~]$ cd Documents/project/
[ec2-user@myServer project]$ 
```

Tip: Use `../` to go up a single directory at a time.

For example, if you are in the /home/userA folder, `cd ../` will navigate to /home.

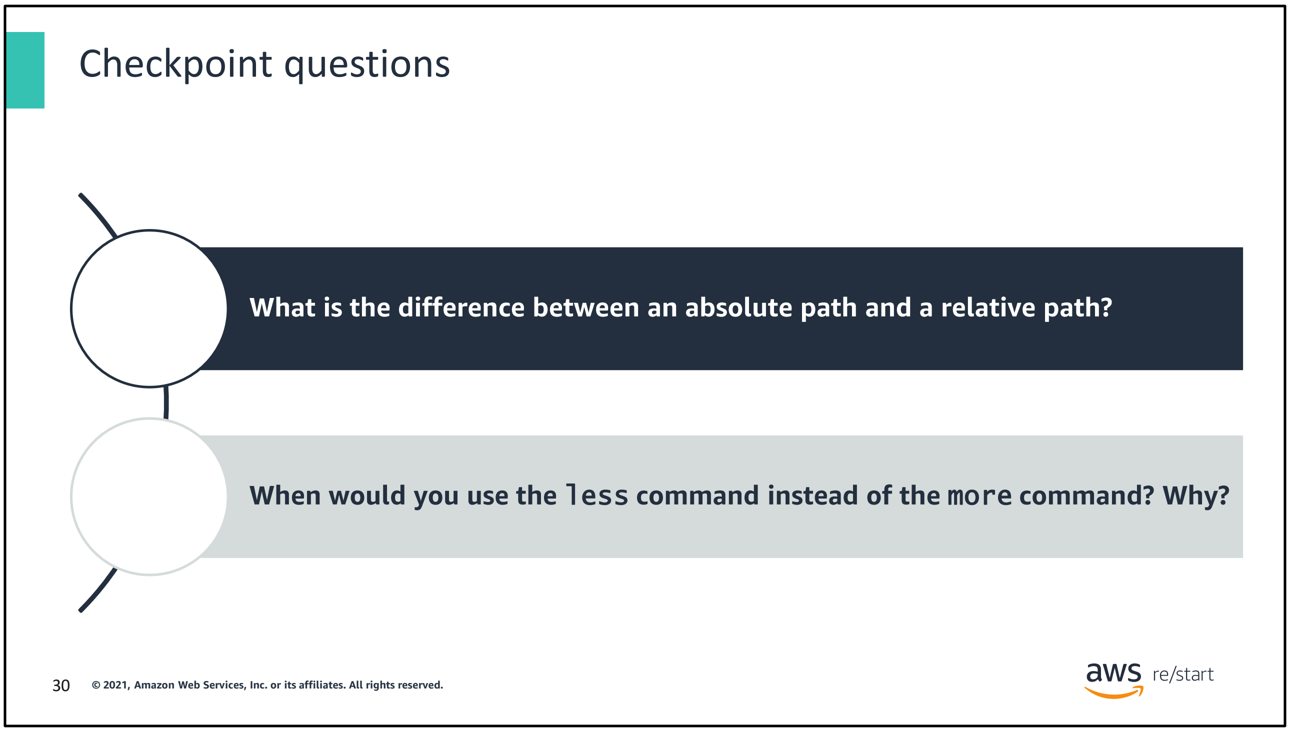
Demonstration: Absolute and relative paths

In this demonstration, the instructor will show you the difference between absolute and relative paths.



```
[ec2-user@myServer CompanyA]$ pwd
/home/ec2-user/CompanyA
[ec2-user@myServer CompanyA]$ cd ..
[ec2-user@myServer ~]$ pwd
/home/ec2-user
[ec2-user@myServer ~]$ cd CompanyA/HR/
[ec2-user@myServer HR]$ pwd
/home/ec2-user/CompanyA/HR
[ec2-user@myServer HR]$ 
```

Follow along as the instructor demonstrates the difference between absolute and relative paths.



Checkpoint questions

What is the difference between an absolute path and a relative path?

When would you use the `less` command instead of the `more` command? Why?

aws re/start

Answers:

1. The absolute path shows the entire folder structure to the resource that is being used. The absolute path to `my_file.txt` in the `Documents` directory would be something like `/Users/user_name/Documents/LabWork/my_file.txt`. The relative path shows only from the current directory to the file that is being used. From the previous example, within the `user_name` directory, the relative path to the file is `/Documents/LabWork/my_file.txt`.
2. You use the `less` command if you want to scroll backward through a file. With the `more` command, you can only scroll forward through a file.

Key takeaways

- Everything in Linux is a file.
- The Linux file system:
  - Is case sensitive
  - Has the key-like directories:
    - `/`
    - `/home`
    - `/mnt`
- Linux contains many commands to help work with files. Some are:
  - `ls` – Lists the contents of a directory
  - `cat` – Shows the contents of a file
  - `cp` – Copies a file
  - `rm` – Removes a file
  - `mkdir` – Creates a directory
- Linux has both absolute and relative directory paths.

Some key takeaways from this lesson include the following:
- Everything in Linux is a file.
- The Linux file system is case sensitive and has key-like directories.
- Linux contains many commands to help work with files.
- Linux has both absolute and relative directory paths.

Thank you

aws re/start

Thank you.