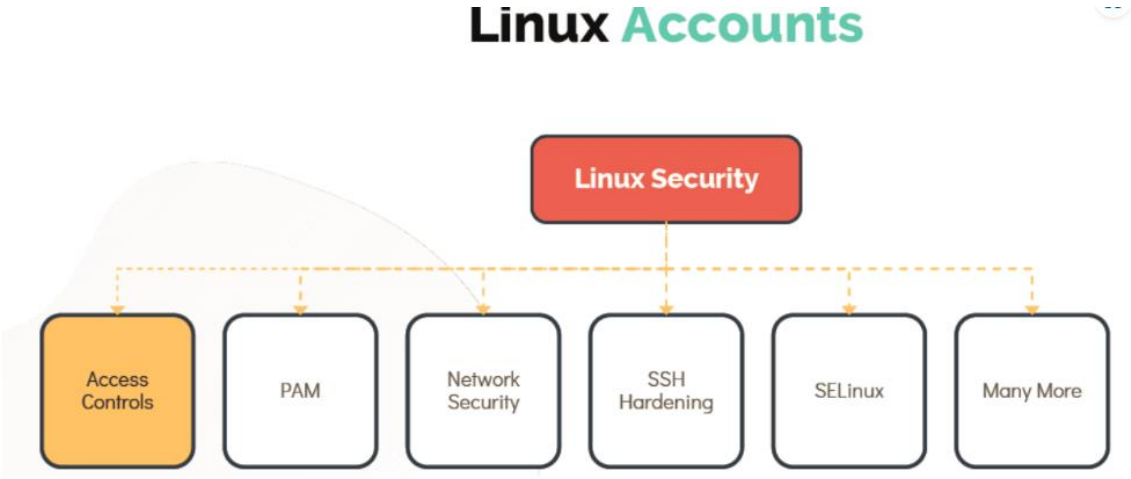
Security-and-File-Permissions

LINUX ACCOUNTS



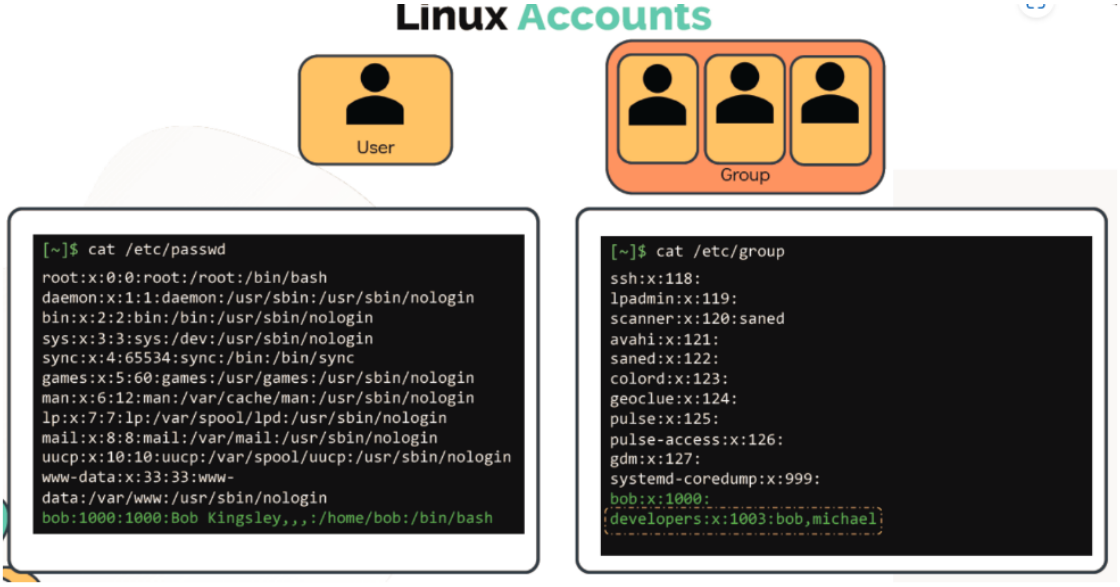
User Accounts

- User's informations are stored under `/etc/passwd` file.

```
[~]$ cat /etc/passwd
```

- Information about groups is stored into `/etc/group` file.

```
[~]$ cat /etc/group
```



- Each user has a username and a unique ID assigned to them known as user ID or UID.
- The user also has a GID, the group id they are part of, `id` command can be use to check these details. for eg:

```
[~]$ id michael
uid=1001(michael) gid=1001(michael)groups=1001(michael),1003(developers)
```

- More details about the user account can be found eg. default shell, home directory using.

```
[~]$ grep -i michael /etc/passwd
michael:x:1001:1001::/home/michael:/bin/sh
```

- To see the list of users currently logged use `who` command.

```
[~]$ who
bob pts/2 Apr 28 06:48 (172.16.238.187)
```

- The `last` command displays the record of all logged-in users along with the date and time when the system was rebooted.

```
[~]$ last
michael :1 :1 Tue May 12 20:00 still logged in
sarah :1 :1 Tue May 12 12:00 still running
reboot system boot 5.3.0-758-gen Mon May 11 13:00 - 19:00 (06:00)
```

Switching users

- To switch to any user use `su` command.

```
[~]$ su -
Password:

root ~#
```

- To run a specific command you can use `su -c "whoami"` (This is not recommended way)

```
[michael@ubuntu-server ~]$ su -c "whoami"
Password:
root
```

- To run a command as a root user `sudo` command is recommended.

```
[michael@ubuntu-server ~]$ sudo apt-get install nginx
[sudo] password for michael:
```

```
[~]$ su -
  Password:
root ~#
```

```
[michael@ubuntu-server ~]$ su -c "whoami"
  Password:
root
```

```
[michael@ubuntu-server ~]$ sudo apt-get install nginx
  [sudo] password for michael:
```

- Users listed in /etc/sudoers file can make use of sudo command for privledge escalation.

```
[~]$ cat /etc/sudoers
```

SUDO

```
[~]$ cat /etc/sudoers
User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root
privileges
%admin ALL=(ALL) ALL
# Allow members of group sudo to execute any
command
%sudo    ALL=(ALL:ALL) ALL
# Allow Bob to run any command
bob ALL=(ALL:ALL) ALL
# Allow Sarah to reboot the system
sarah localhost=/usr/bin/shutdown -r now
# See sudoers(5) for more information on "#include"
directives:
#includedir /etc/sudoers.d
```

Field	Description	Example
1	User or Group	bob, %sudo (group)
2	Hosts	localhost, ALL(default)
3	User	ALL(default)
4	Command	/bin/ls, ALL(unrestricted)

- To restrict anyone from directly login as root login, this can be done by setting `nologin` shell.

```
[~]$ grep -i ^root /etc/passwd
/root:x:0:0:root:/root:/usr/sbin/nologin
```

USER MANAGEMENT

User Add

- To create a new local user `bob` in the system use `useradd` command.

```
[~]$ useradd bob
```

- To get more details about `bob` account like, home director, uid, and shell use `/etc/passwd`

```
[~]$ grep -i bob /etc/passwd
bob:x:1002:1002::/home/bob:/bin/sh
```

- To check the uid or username of the user logged in user `whoami` command.

```
[~]$ whoami
bob
```

- All user's password are store under `/etc/shadow`

```
[~]$ grep -i bob /etc/shadow
bob:!:18341:0:99999:7:::
```

- To change the password of current user use `passwd` or for any specific user use `passwd <username>`

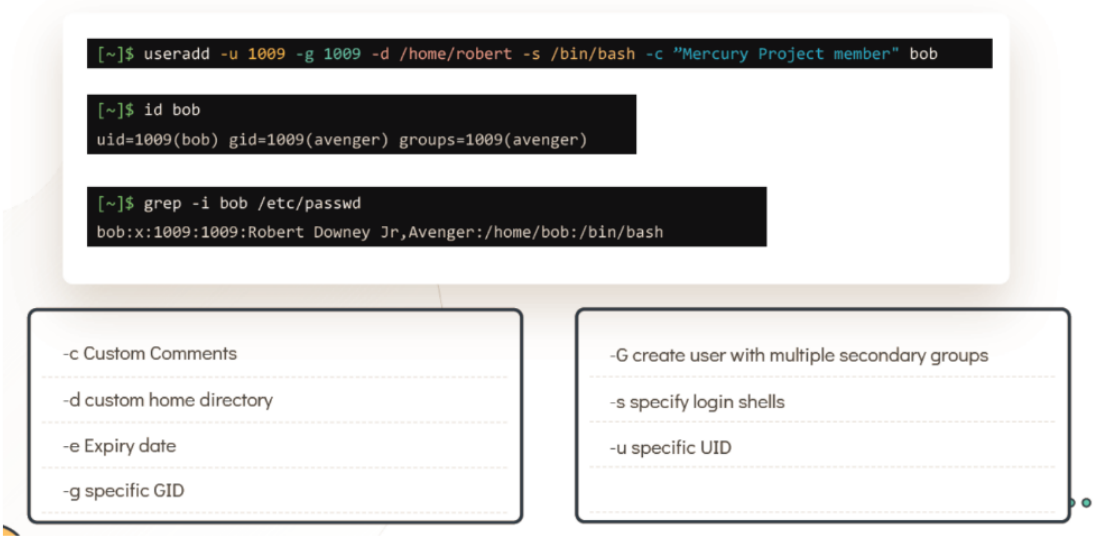
```
[~]$ passwd bob
Changing password for user bob.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated
successfully.
```

Managing Users

- `useradd` command be used along with many attributes as show below.

```
[~]$ useradd -u 1009 -g 1009 -d /home/robert -s /bin/bash -c "Mercury Project member" bob
```

Managing Users



- To delete a user use `userdel` command

```
[~]$ userdel bob
```

- To add a group use `groupadd` command

```
[~]$ groupadd -g 1011 developer
```

- To delete a group user `groupdel` command

```
[~]$ groupdel developer
```

ACCESS CONTROL FILES

- Access Ccontrol files are stored under `/etc`.
- Can be read by anyone and can be only edited by `root` user.

Control files

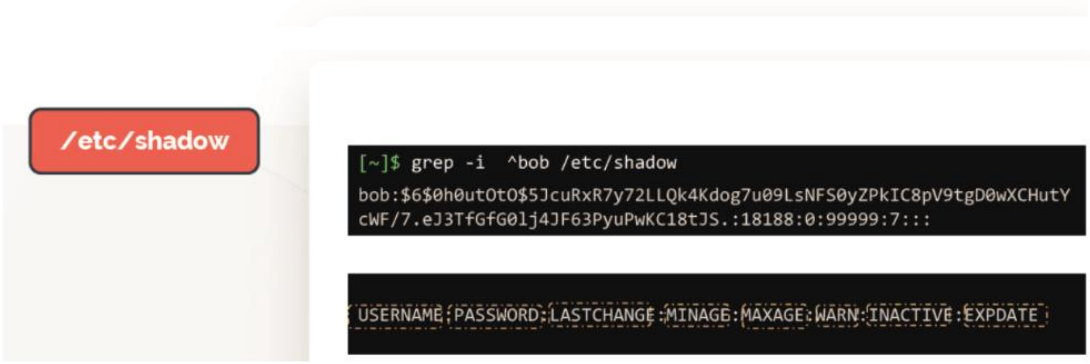
- To get more details about one's account for example `bob` account, home director, uid, and shell check `/etc/passwd`

```
[~]$ grep -i ^bob /etc/passwd
bob:x:1002:1002::/home/bob:/bin/sh
USERNAME:PASSWORD:UID:GID:GECOS:HOMEDIR:SHELL
```

- Password are stored under `/etc/shadow`

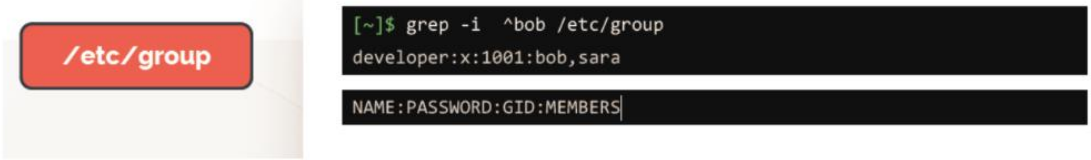
```
[~]$ grep -i ^bob /etc/shadow
bob:$6$0h0utOtO$5JcuRxR7y72LLQk4Kdog7u09LsNFS0yZPkIC8pV9tgD0wXCHutY
cWF/7.eJ3TfGfG0lj4JF63PyuPwKC18tJS.:18188:0:99999:7:::

USERNAME:PASSWORD:LASTCHANGE:MINAGE:MAXAGE:WARN:INACTIVE:EXPDATE
```

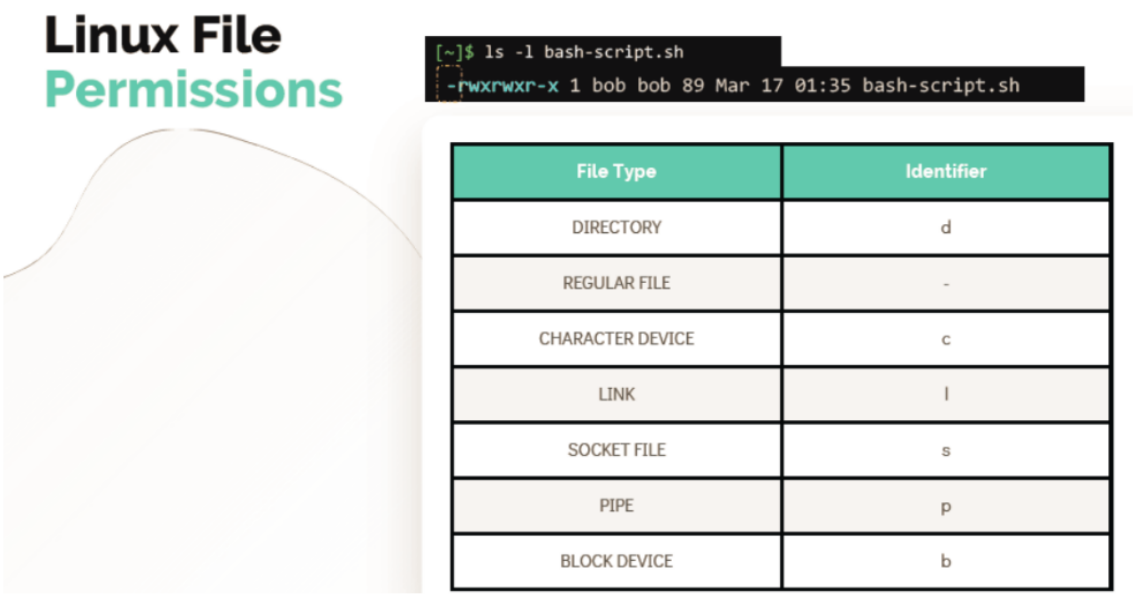


- Check the groups `bob` belongs too

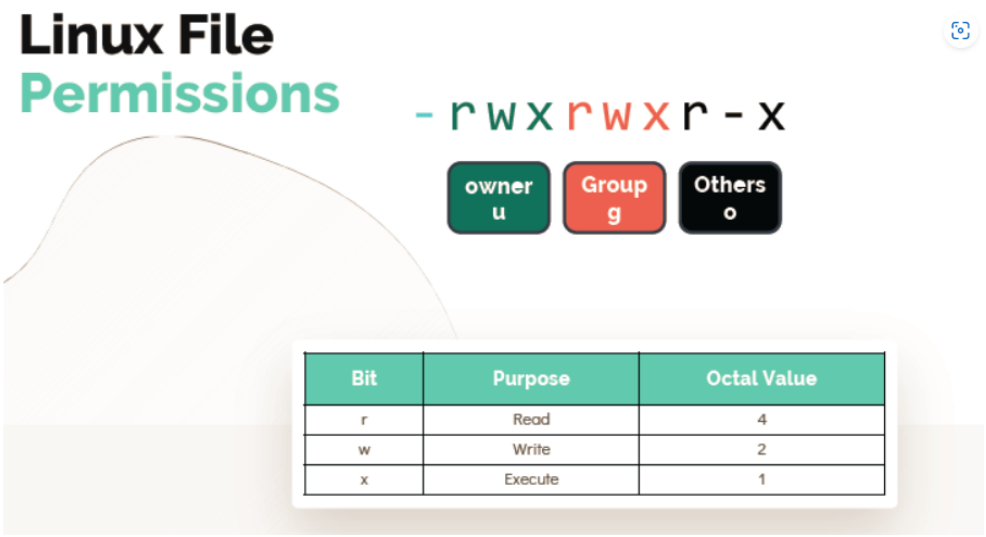
```
[~]$ grep -i ^bob /etc/group
NAME:PASSWORD:GID:MEMBERS
```



LINUX FILE PERMISSIONS



File Type	Identifier
DIRECTORY	d
REGULAR FILE	-
CHARACTER DEVICE	c
LINK	l
SOCKET FILE	s
PIPE	p
BLOCK DEVICE	b



Bit	Purpose	Octal Value
r	Read	4
w	Write	2
x	Execute	1

Directory Permission

- To list the directory permission use

```
[~]$ ls -ld /home/bob/random_dir
```

- To know the current user

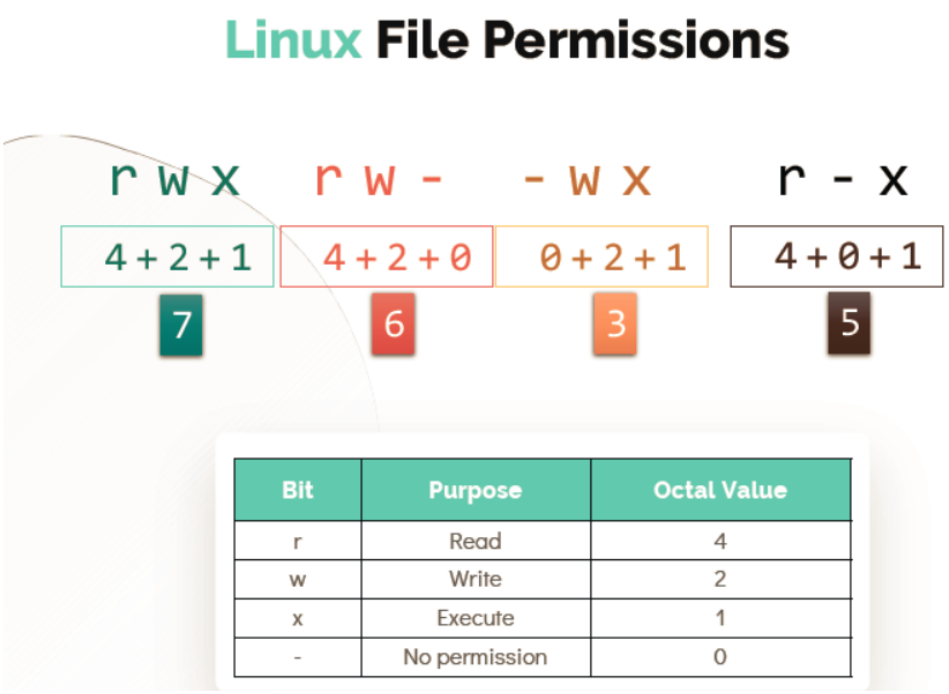
```
[~]$ whoami
```

- To change the change the directory

```
[~]$ cd /home/bob/random_dir
```

File Permissions

- Linux file permissions are defined as



Bit	Purpose	Octal Value
r	Read	4
w	Write	2
x	Execute	1
-	No permission	0

Modifying file permissions

- Use `chmod` command to modify the file permissions.
- Provide full access to owners

```
[~]$ chmod u+rwx test-file
```

- Provide Read access to Owners, groups and others, Remove execute access

```
[~]$ chmod ugo+r-x test-file
```

- Remove all access for others

```
[~]$ chmod o-rwx test-file
```

- Full access for Owner, add read , remove execute for group and no access for others

```
[~]$ chmod u+rwx,g+r-x,o-rwx test-file
```

- Provide full access to Owners, group and others

```
[~]$ chmod 777 test-file
```

- Provide Read and execute access to Owners,groups and others

```
[~]$ chmod 777 test-file
```

- Read and Write access for Owner and Group, No access for others.

```
[~]$ chmod 660 test-file
```

- Full access for Owner, read and execute for group and no access for others.

```
[~]$ chmod 750 test-file
```

Change Ownership

- Changes owner to bob and group to developer

```
[~]$ chown bob:developer test-file
```

- Changes just the owner of the file to bob. Group unchanged.

```
[~]$ chown bob andoid.apk
```

- Change the group for the test-file to the group called android.

```
[~]$ chgrp android test-file
```

SSH and SCP

- SSH is used to login to the remote computer.
- SCP is used to copy of files/directories within the file system also can copy data to remote computer.

SSH

- To login to the remote server use `ssh` command with hostname or IP address.

```
ssh <hostname OR IP Address>
```

- To login to the remote server with specific username and password.

```
ssh <user>@<hostname OR IP Address>
```

`-l` attribute can also be used as

```
ssh -l <user> <hostname OR IP Address>
```

Password-Less Authentication

- Passwordless authentication can be setup via key-pair authentication in order to login to the remote server with password.
- Public and Private key are stored at below location.

```
Public Key: /home/bob/.ssh/id_rsa.pub
```

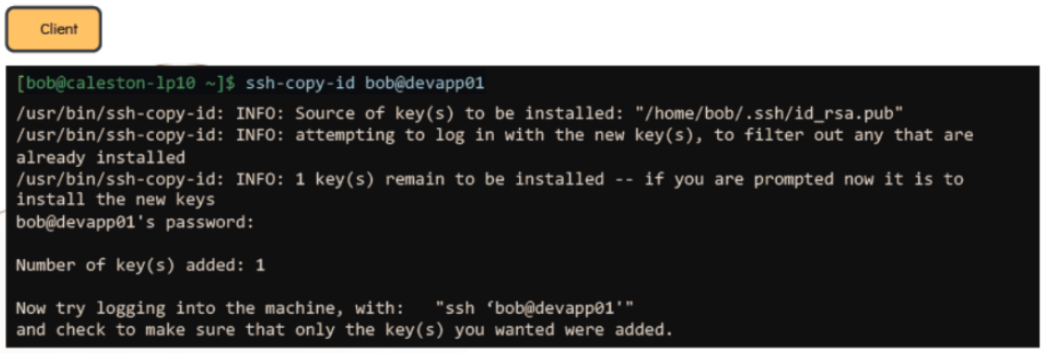
```
Private Key: /home/bob/.ssh/id_rsa
```

- To generate a keypair on the `Client` run this command

```
bob@caleston-lp10 ~]$ ssh-keygen -t rsa
```

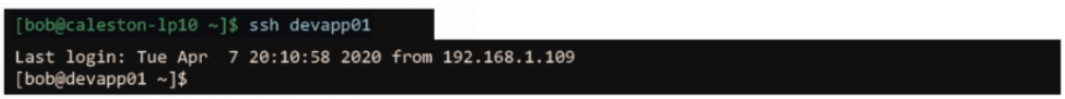
- To copy the Public key from the client to the remote server

```
bob@caleston-lp10 ~]$ ssh-copy-id bob@devapp01
```



- Now `Bob` can login to remote server without password

```
[bob@caleston-lp10 ~]$ ssh devapp01
```



- Public Key is copied to the remote server at :

```
[bob@caleston-lp10 ~]$ cat /home/bob/.ssh/authorized_keys
```

Remote Server

```
[bob@caleston-lp10 ~]$ cat /home/bob/.ssh/authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCgVV5wgH37kNwjnEIxgeX4j6LASNckjKi4bRpjPGecyxEiEeJhIU4x31XPEFzUFp/1xX2rj
eiM2Ko3oPmTGCCTEQMpQogerR7NS+bA9eXs34jWIg+xoSQjeQu1+lXgrRippJn2YhWYVAY3sKWIiiklowuMXmxjmBBr48L52di1J+
8EASwnM4ILX/YL72Czq3uFFhVW1fNUKBPUbW58h4QSAd2r9abzZfrHH48ThPJW4/5i8LOHEo3W0BXl3foEV0c6pk3TgxcjTuZQOim
d48mM2pxWJh9WxA0xcXwbD3+JrcnZeMJq4TbrKjaXQ0pBGenglxurxnRT2og9DeTIqGN3 bob@caleston-lp10
```

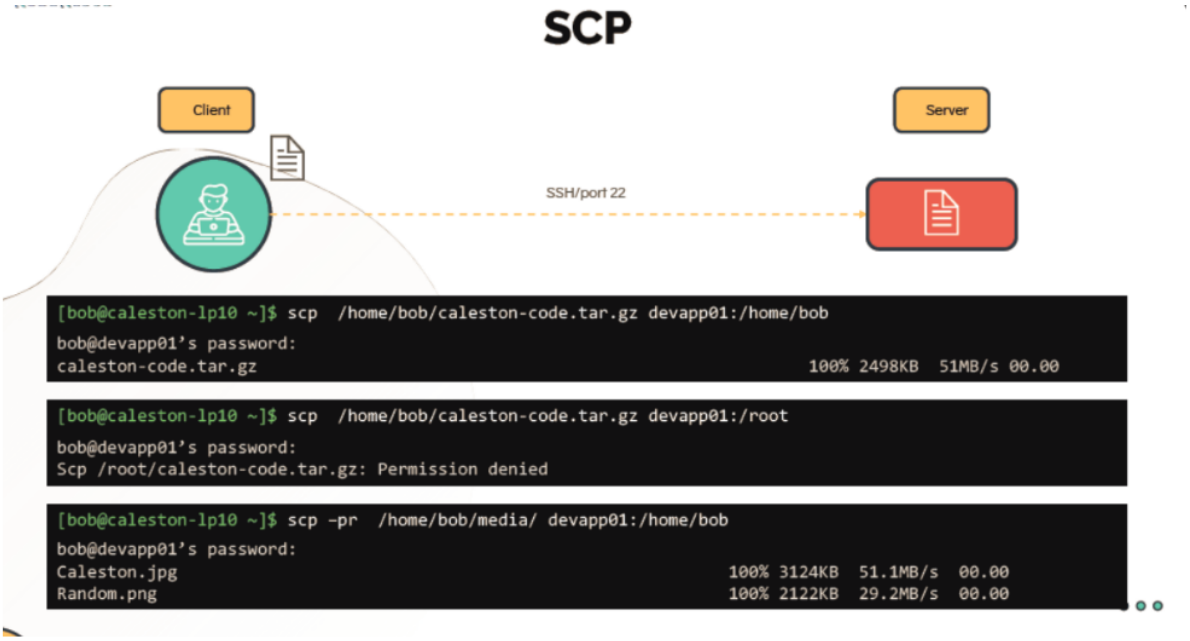
SCP

- To copy a compresses file to a remote server

```
bob@caleston-lp10 ~]$ scp /home/bob/caleston-code.tar.gz devapp01:/home/bob
```

- To copy a directory to a remote server

```
[bob@caleston-lp10 ~]$ scp -pr /home/bob/media/ devapp01:/home/bob
```



Cronjob in Linux

The basic usage of **cron** is to execute a job in a specific time. The `crontab` is a list of commands that you want to run on a regular schedule, and also the name of the command used to manage that list. **Crontab** stands for `cron table` because it uses the job scheduler cron to execute tasks. The schedule is called the crontab, which is also the name of the program used to edit that schedule.

Linux Crontab Format

Format

```
Min  Hour Day  Mon  Weekday

*    *    *    *    *  command to be executed


T    T    T    T    T
|    |    |    |    └─  Day of Week   (0=Sun .. 6=Sat)
|    |    |    └──────  Month         (1..12)
|    |    └───────────  Day of Month  (1..31)
|    └────────────────  Hour          (0..23)
└─────────────────────  Minute        (0..59)
```

Field	Range	Special characters
Minute	0 - 59	, - * /
Hour	0 - 23	, - * /
Day of Month	1 - 31	, - * ? / L W
Month	1 - 12	, - * /
Day of Week	0 - 6	, - * ? / L #

Expressions used and Description

Special strings

@reboot	Run once, at system startup (non-standard)
@yearly	Run once every year, "0 0 1 1 *" (non-standard)
@annually	(same as @yearly) (non-standard)
@monthly	Run once every month, "0 0 1 * *" (non-standard)
@weekly	Run once every week, "0 0 * * 0" (non-standard)
@daily	Run once each day, "0 0 * * *" (non-standard)
@midnight	(same as @daily) (non-standard)
@hourly	Run once an hour, "0 * * * *" (non-standard)

Special characters	
Asterik(*)	Matches all values in the field or any possible value.
Hyphen(-)	Used to define a range.Ex: 1-5 in 5th field(Day Of Week) Every Weekday i.e., Monday to Friday
Slash (/)	1st field(Minute) /15 meaning every fifteen minute or increment of range.
Comma (,)	Used to separate items.Ex: 2,6,8 in 2nd fields(Hour) executes at 2am,6am and 8am
L	It is allowed only for Day of Month or Day Of Week field, 2L in Day of week indicates Last tuesday of every month
Hash (#)	It is allowed only for Day Of Week field, which must be followed within range of 1 to 5. For example, 4#1 means "The first Thursday" of given month.
Question mark (?)	Can be instead of '*' and allowed for Day of Month and Day Of Week. Usage is restricted to either Day of Month or Day Of Week in a cron expression.

Crontab commands

Crontab command	
`crontab -e`	Edit or create a crontab file if doesn't already exist.
`crontab -l`	Display the crontab file.
`crontab -r`	Remove the crontab file.
`crontab -v`	Display the last time you edited your crontab file. (non-standard)

Crontab Examples

*/30 * * * * Every 30 mins

0 * * * * Every hour

0 0 * * 0 At midnight of every Sunday

0 0 0 15 * * Every 15th of month (monthly)

0 0 0 1 1 * Every 1st of january (yearly)

@reboot Every reboot