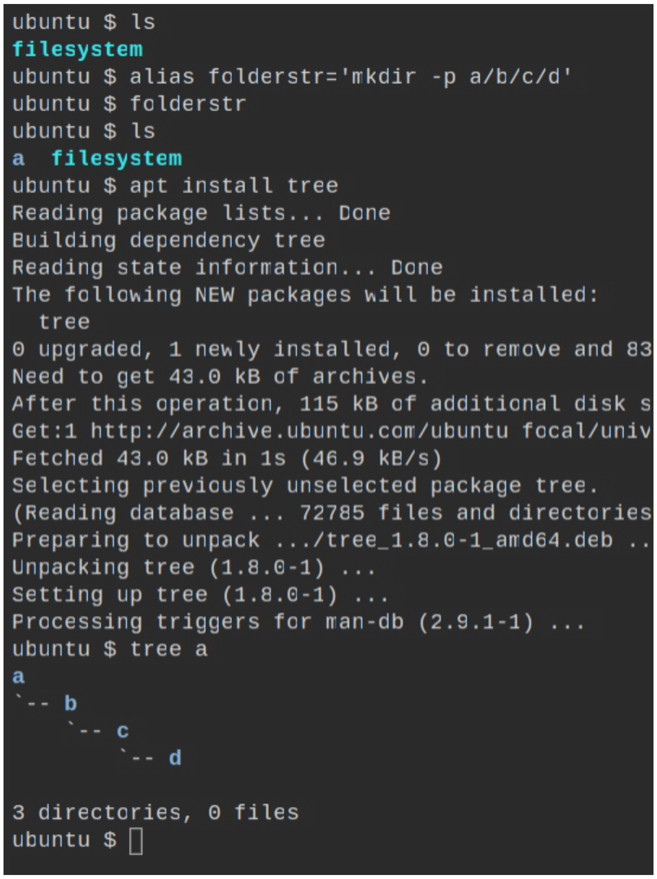
Linux Day 03 - Assignment 💻

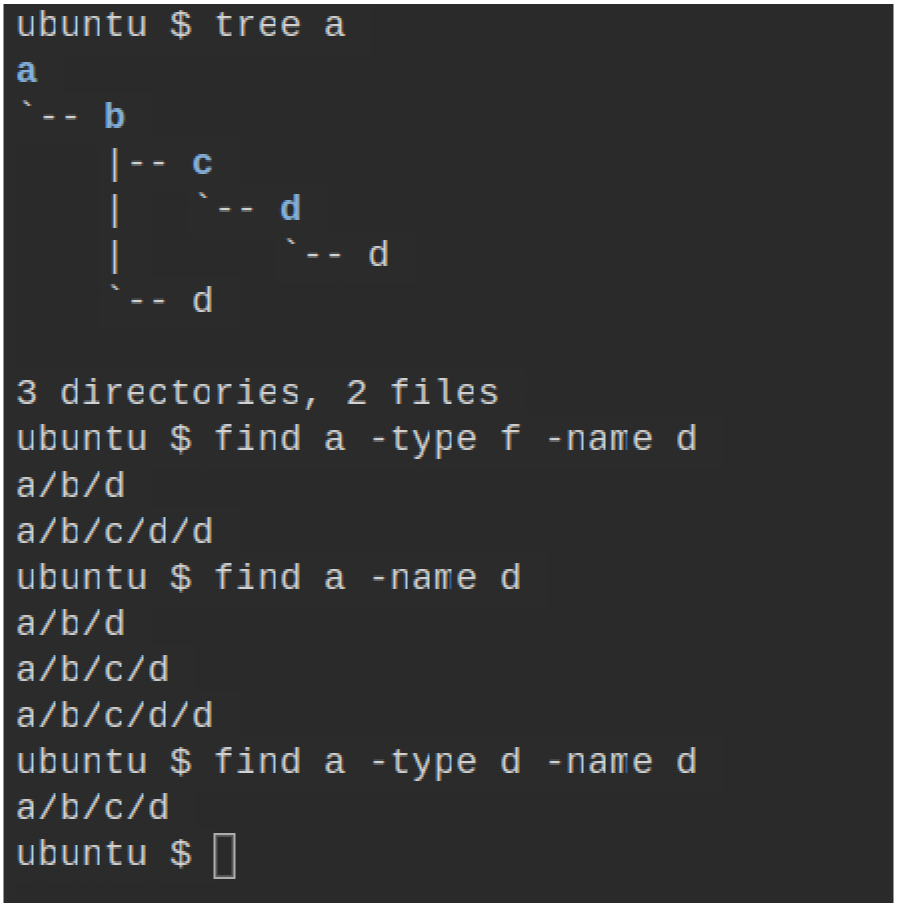
Create an alias for mkdir -p a/b/c/d

alias folderstr='mkdir -p a/b/c/d



```
ubuntu $ ls
filesystem
ubuntu $ alias folderstr='mkdir -p a/b/c/d'
ubuntu $ folderstr
ubuntu $ ls
a  filesystem
ubuntu $ apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 83
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk s
Get:1 http://archive.ubuntu.com/ubuntu focal/univ
Fetched 43.0 kB in 1s (46.9 kB/s)
Selecting previously unselected package tree.
(Reading database ... 72785 files and directories
Preparing to unpack .../tree_1.8.0-1_amd64.deb ..
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu $ tree a
a
`-- b
    `-- c
        `-- d

3 directories, 0 files
ubuntu $ 
```

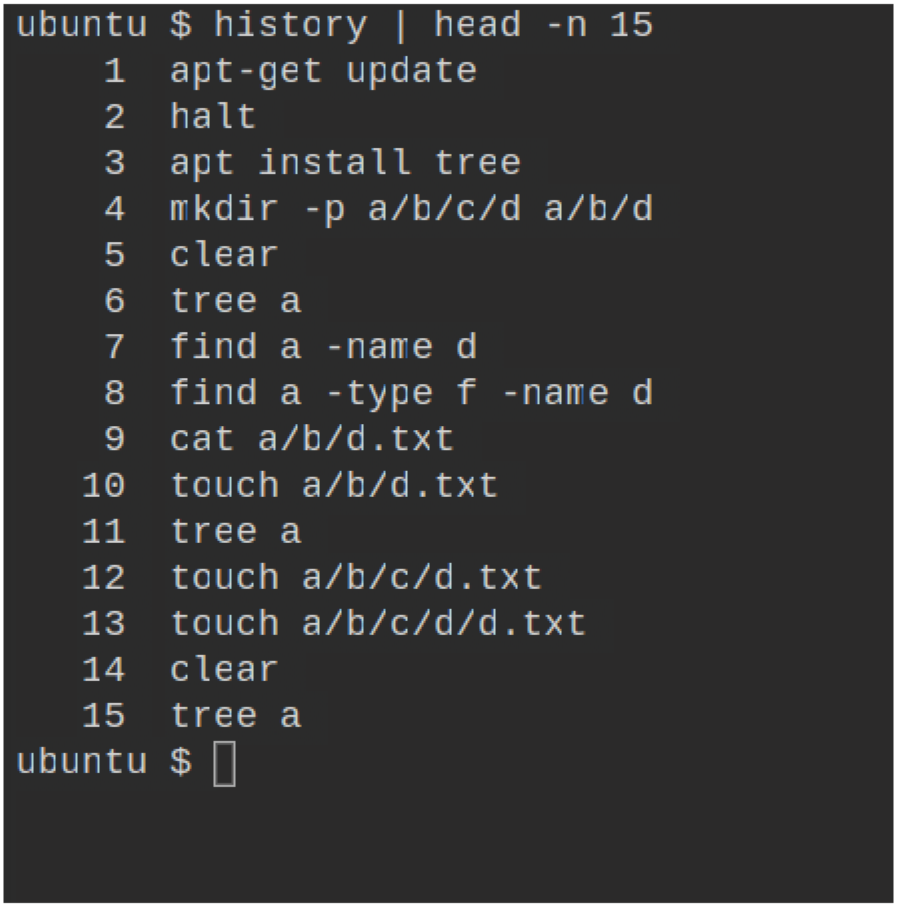
How do you search for a file in linux ? (You can create one in a subdirectory and test)



```
ubuntu $ tree a
a
`-- b
    |-- c
    |   `-- d
    |       `-- d
    `-- d

3 directories, 2 files
ubuntu $ find a -type f -name d
a/b/d
a/b/c/d/d
ubuntu $ find a -name d
a/b/d
a/b/c/d
a/b/c/d/d
ubuntu $ find a -type d -name d
a/b/c/d
ubuntu $ 
```

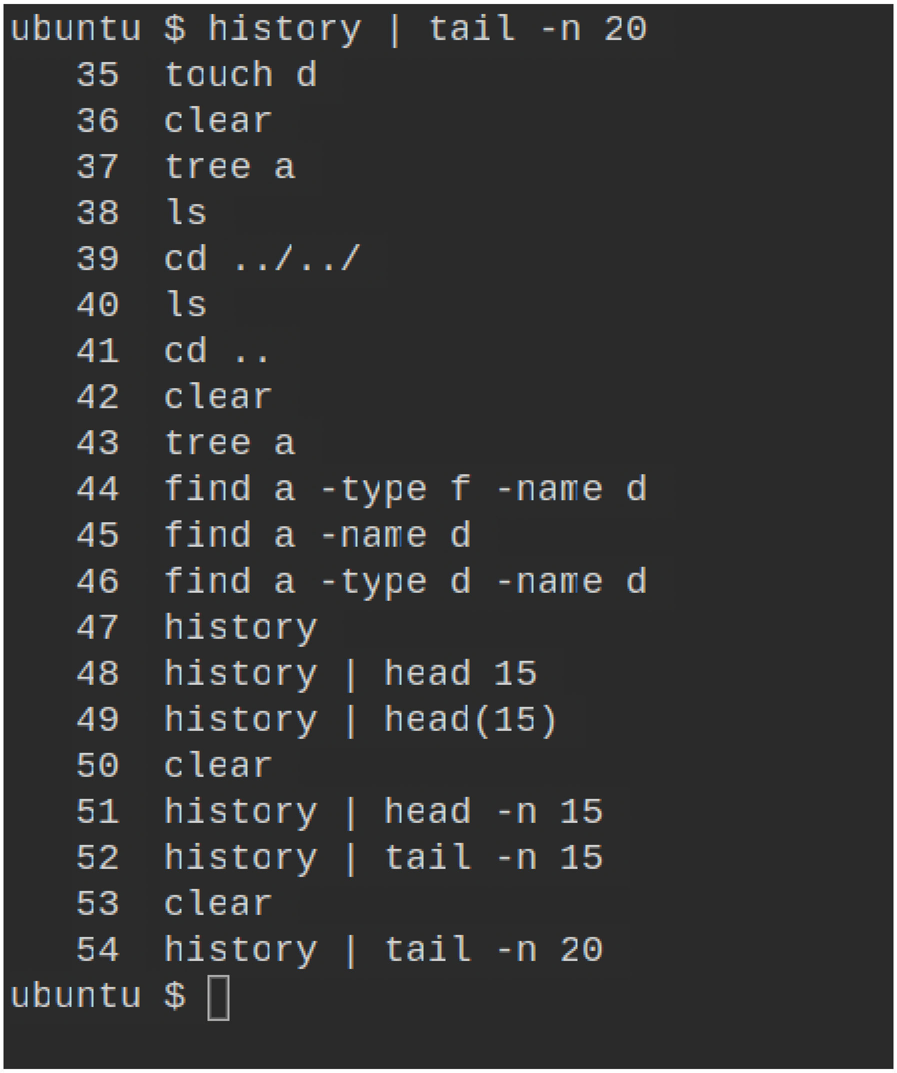
Link for reference:
https://www.freecodecamp.org/news/how-to-search-for-files-from-the-linux-command-line/

How to print only the first 15 lines of the output of a command(history) ?



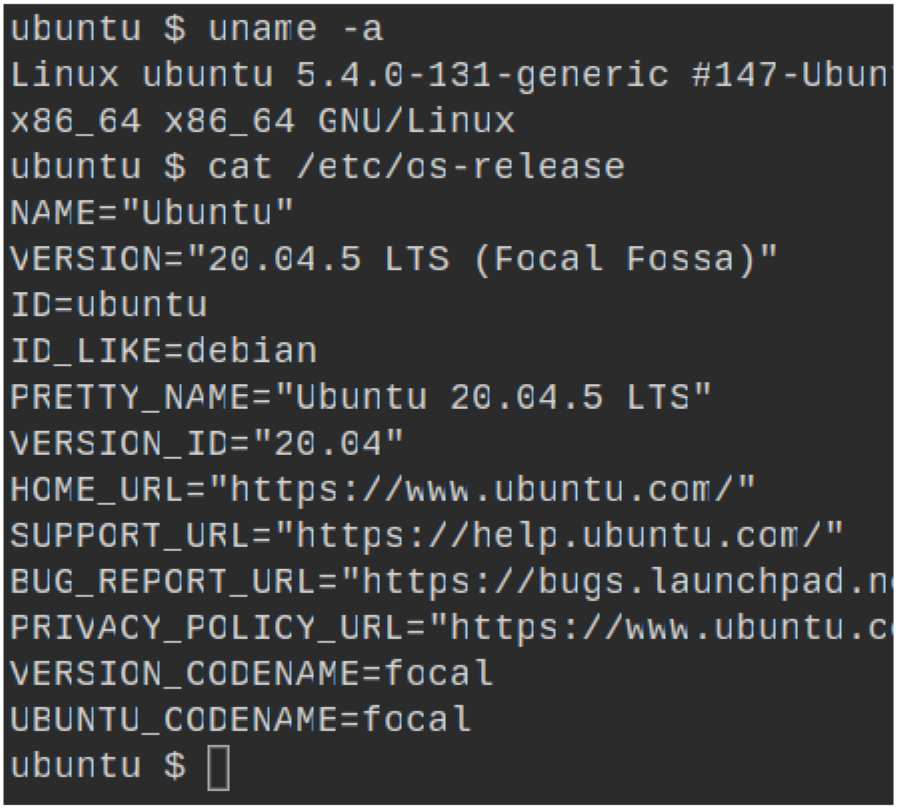
```
ubuntu $ history | head -n 15
    1  apt-get update
    2  halt
    3  apt install tree
    4  mkdir -p a/b/c/d a/b/d
    5  clear
    6  tree a
    7  find a -name d
    8  find a -type f -name d
    9  cat a/b/d.txt
   10  touch a/b/d.txt
   11  tree a
   12  touch a/b/c/d.txt
   13  touch a/b/c/d/d.txt
   14  clear
   15  tree a
ubuntu $ 
```

How to print the last 20 lines of the ouput of a command(history) ?



```
ubuntu $ history | tail -n 20
   35  touch d
   36  clear
   37  tree a
   38  ls
   39  cd ../../
   40  ls
   41  cd ..
   42  clear
   43  tree a
   44  find a -type f -name d
   45  find a -name d
   46  find a -type d -name d
   47  history
   48  history | head 15
   49  history | head(15)
   50  clear
   51  history | head -n 15
   52  history | tail -n 15
   53  clear
   54  history | tail -n 20
ubuntu $
```

How to check the OS related information of your linux machine ? (Distribution, Arch etc)



```
ubuntu $ uname -a
Linux ubuntu 5.4.0-131-generic #147-Ubun
x86_64 x86_64 GNU/Linux
ubuntu $ cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.5 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.5 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.n
PRIVACY_POLICY_URL="https://www.ubuntu.c
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
ubuntu $ 
```

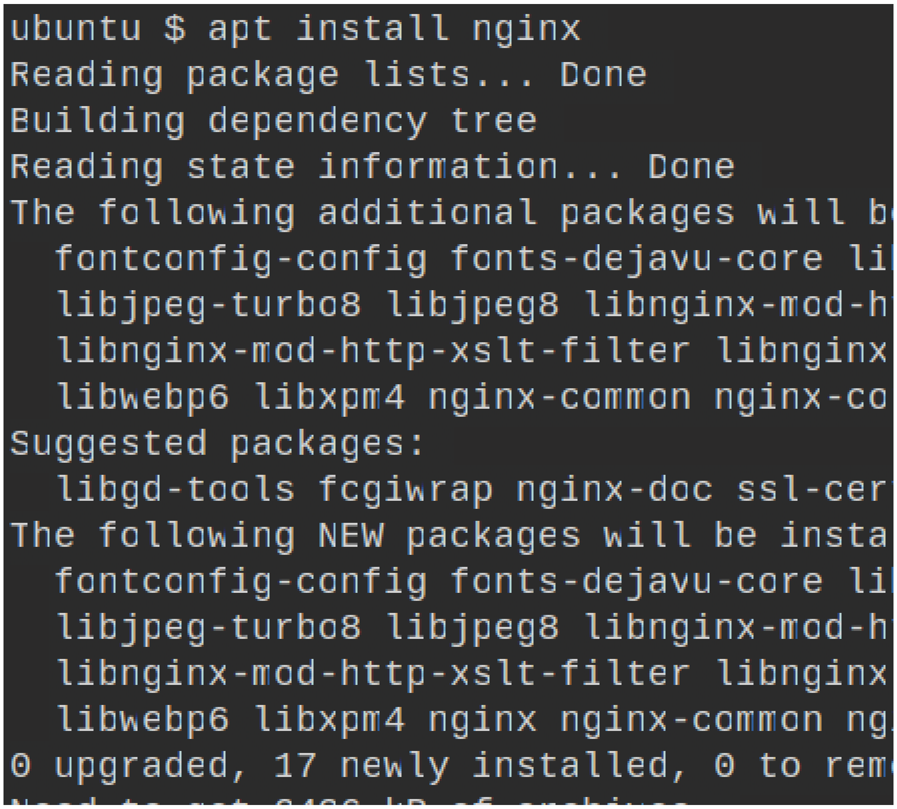
What is a webserver ? [Explain]

A web server is a computer program that processes requests from clients (typically web browsers) and sends them the appropriate response, typically in the form of web pages, images, or other content. In other words, a web server is a software application that serves as the backbone of the World Wide Web, making it possible to access and interact with websites and web-based applications.

When a user requests a webpage or resource, the web server processes that request and returns the corresponding content to the user's browser. The content may be generated dynamically from a database or other data source, or it may be pre-existing files such as HTML, CSS, and image files that are stored on the server.

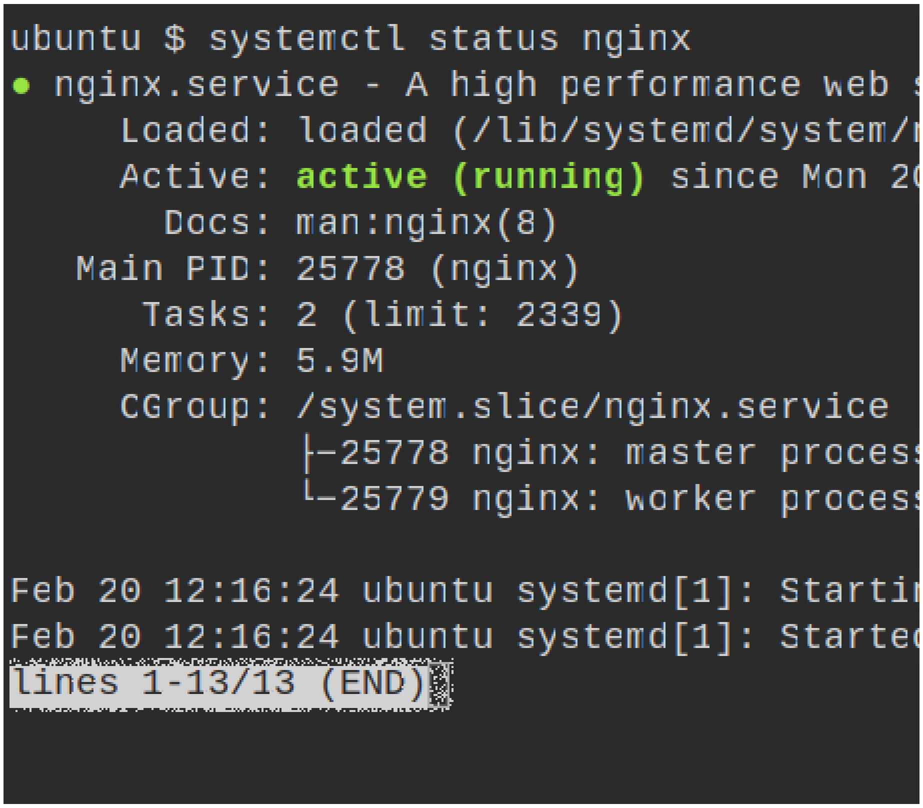
Some examples of web servers include Apache, Nginx, and Microsoft's Internet Information Services (IIS). These servers run on powerful computers that are connected to the internet, and they can serve a large number of requests simultaneously, making it possible for millions of users to access websites and web applications at the same time.

Install nginx on your linux machine with apt



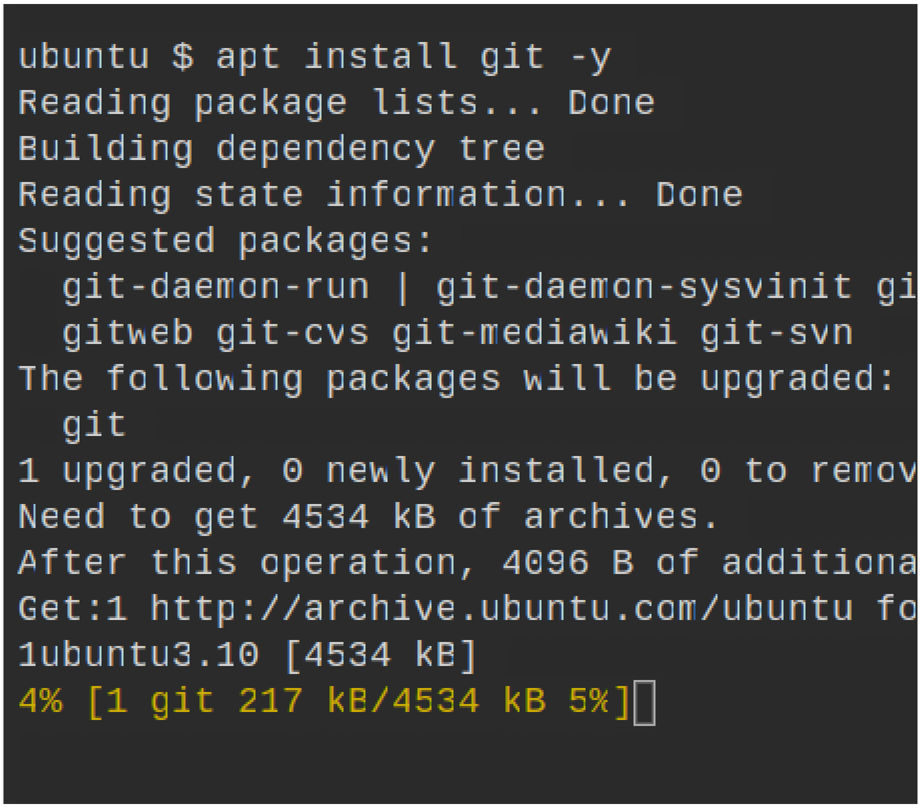
```
ubuntu $ apt install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will b
  fontconfig-config fonts-dejavu-core li
  libjpeg-turbo8 libjpeg8 libnginx-mod-h
  libnginx-mod-http-xslt-filter libnginx
  libwebp6 libxpm4 nginx-common nginx-co
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cer
The following NEW packages will be insta
  fontconfig-config fonts-dejavu-core li
  libjpeg-turbo8 libjpeg8 libnginx-mod-h
  libnginx-mod-http-xslt-filter libnginx
  libwebp6 libxpm4 nginx nginx-common ng
0 upgraded, 17 newly installed, 0 to rem
```

Verify if nginx is up and running



```
ubuntu $ systemctl status nginx
● nginx.service - A high performance web s
     Loaded: loaded (/lib/systemd/system/n
     Active: active (running) since Mon 20
       Docs: man:nginx(8)
   Main PID: 25778 (nginx)
      Tasks: 2 (limit: 2339)
     Memory: 5.9M
     CGroup: /system.slice/nginx.service
             ├─25778 nginx: master process
             └─25779 nginx: worker process

Feb 20 12:16:24 ubuntu systemd[1]: Startin
Feb 20 12:16:24 ubuntu systemd[1]: Started
lines 1-13/13 (END)
```

Install git using apt

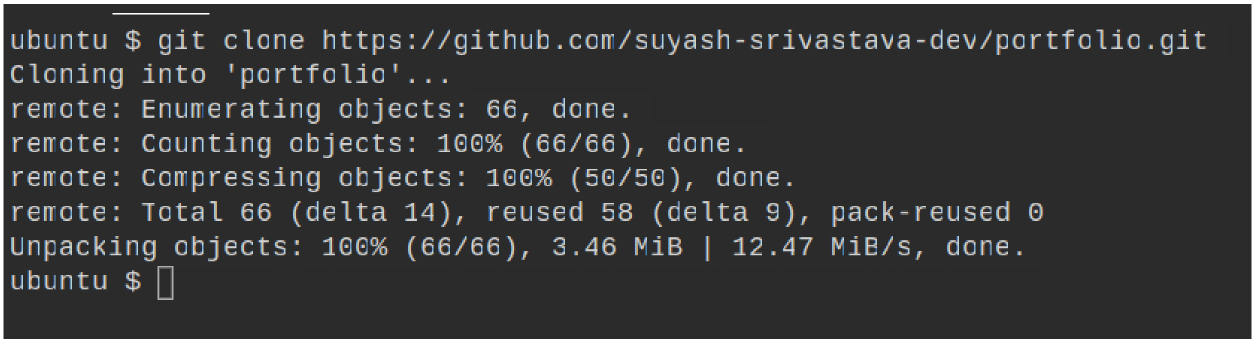


```
ubuntu $ apt install git -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  git-daemon-run | git-daemon-sysvinit gi
  gitweb git-cvs git-mediawiki git-svn
The following packages will be upgraded:
  git
1 upgraded, 0 newly installed, 0 to remov
Need to get 4534 kB of archives.
After this operation, 4096 B of additiona
Get:1 http://archive.ubuntu.com/ubuntu fo
1ubuntu3.10 [4534 kB]
4% [1 git 217 kB/4534 kB 5%]
```

Deploy the portfolio via Nginx and share a screenshot of the webpage as answer

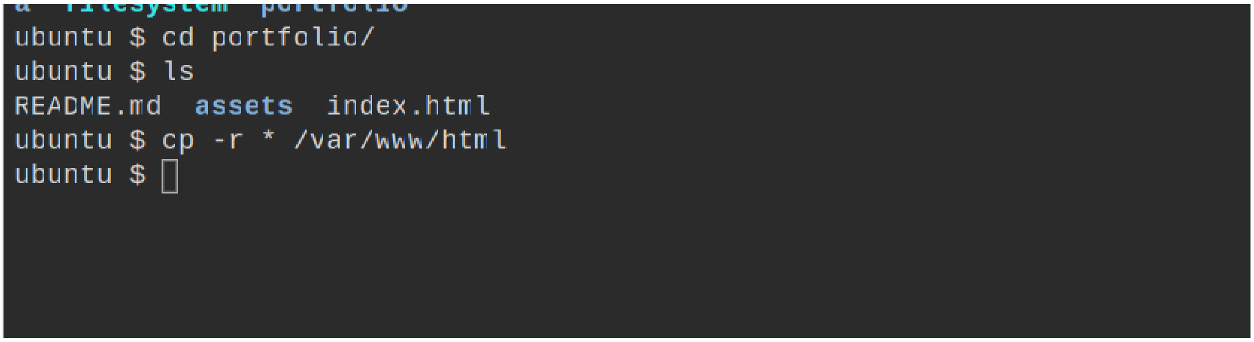
https://github.com/sankalp7654/simple-portfolio

1. Clone repo:



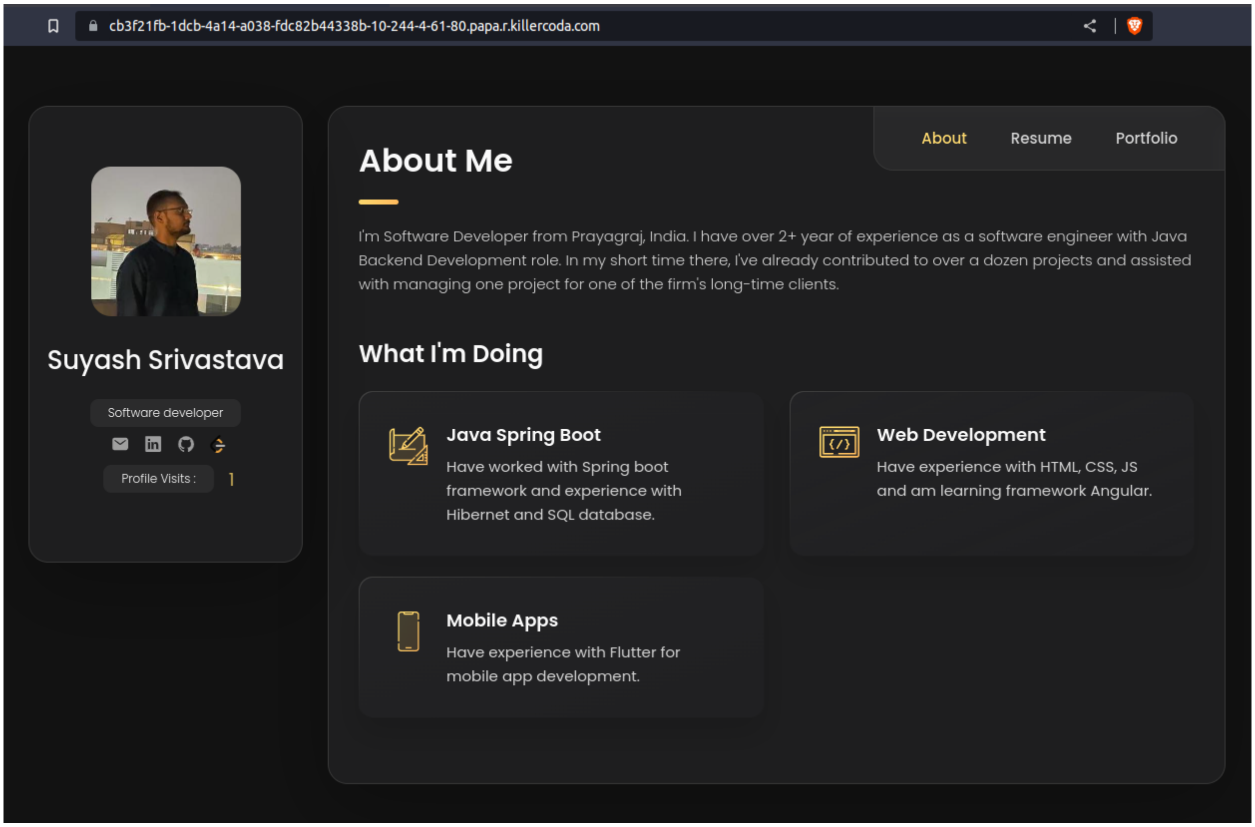
```
ubuntu $ git clone https://github.com/suyash-srivastava-dev/portfolio.git
Cloning into 'portfolio'...
remote: Enumerating objects: 66, done.
remote: Counting objects: 100% (66/66), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 66 (delta 14), reused 58 (delta 9), pack-reused 0
Unpacking objects: 100% (66/66), 3.46 MiB | 12.47 MiB/s, done.
ubuntu $
```

2. Move to directory for html file, copy the file content and expose to nginx



```
a  filesystem  portfolio
ubuntu $ cd portfolio/
ubuntu $ ls
README.md  assets  index.html
ubuntu $ cp -r * /var/www/html
ubuntu $ 
```

3. Open the machine address + port 80



cb3f21fb-1dcb-4a14-a038-fdc82b44338b-10-244-4-61-80.papa.r.killercoda.com

About Me

I'm Software Developer from Prayagraj, India. I have over 2+ year of experience as a software engineer with Java Backend Development role. In my short time there, I've already contributed to over a dozen projects and assisted with managing one project for one of the firm's long-time clients.

What I'm Doing

Java Spring Boot
Have worked with Spring boot framework and experience with Hibernet and SQL database.

Web Development
Have experience with HTML, CSS, JS and am learning framework Angular.

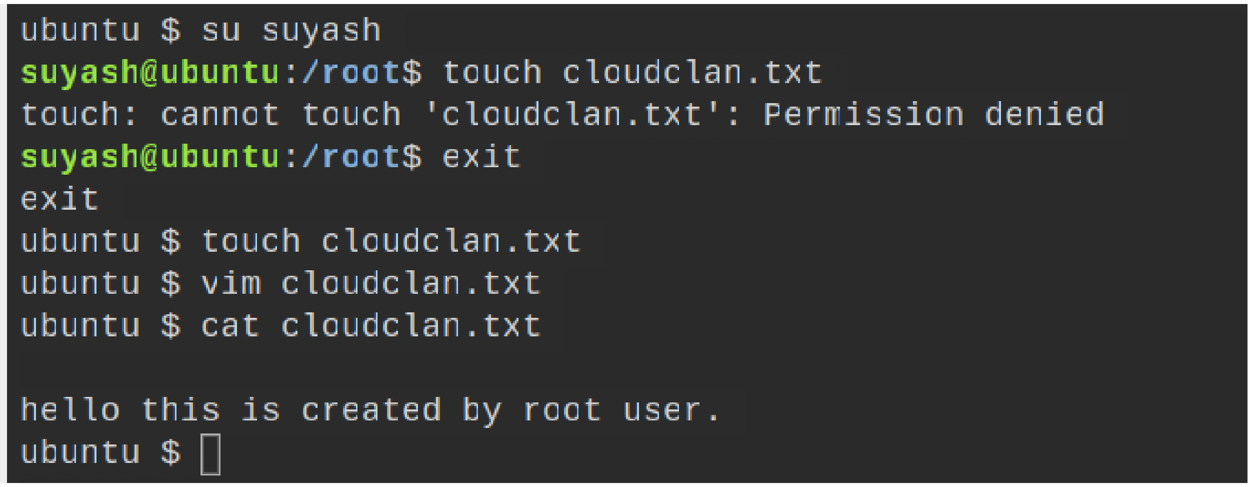
Mobile Apps
Have experience with Flutter for mobile app development.

What is a primary group in linux ? [Explain]

A primary group is the main group associated with a user account. Every user account in the system must be associated with a primary group. The primary group is specified in the user's account entry in the system's password file (usually /etc/passwd) and is identified by a group ID (GID). By default, the primary group for a new user is a group with the same name as the user's username, but this can be changed by the system administrator. A user's primary group determines the default permissions for files and directories that the user creates. When a user creates a new file or directory, the group ownership is set to the user's primary group. This means that members of the user's primary group have the same permissions as the owner of the file or directory. In addition to the primary group, a user can also be a member of multiple secondary groups, which allows for more fine-grained permission management.

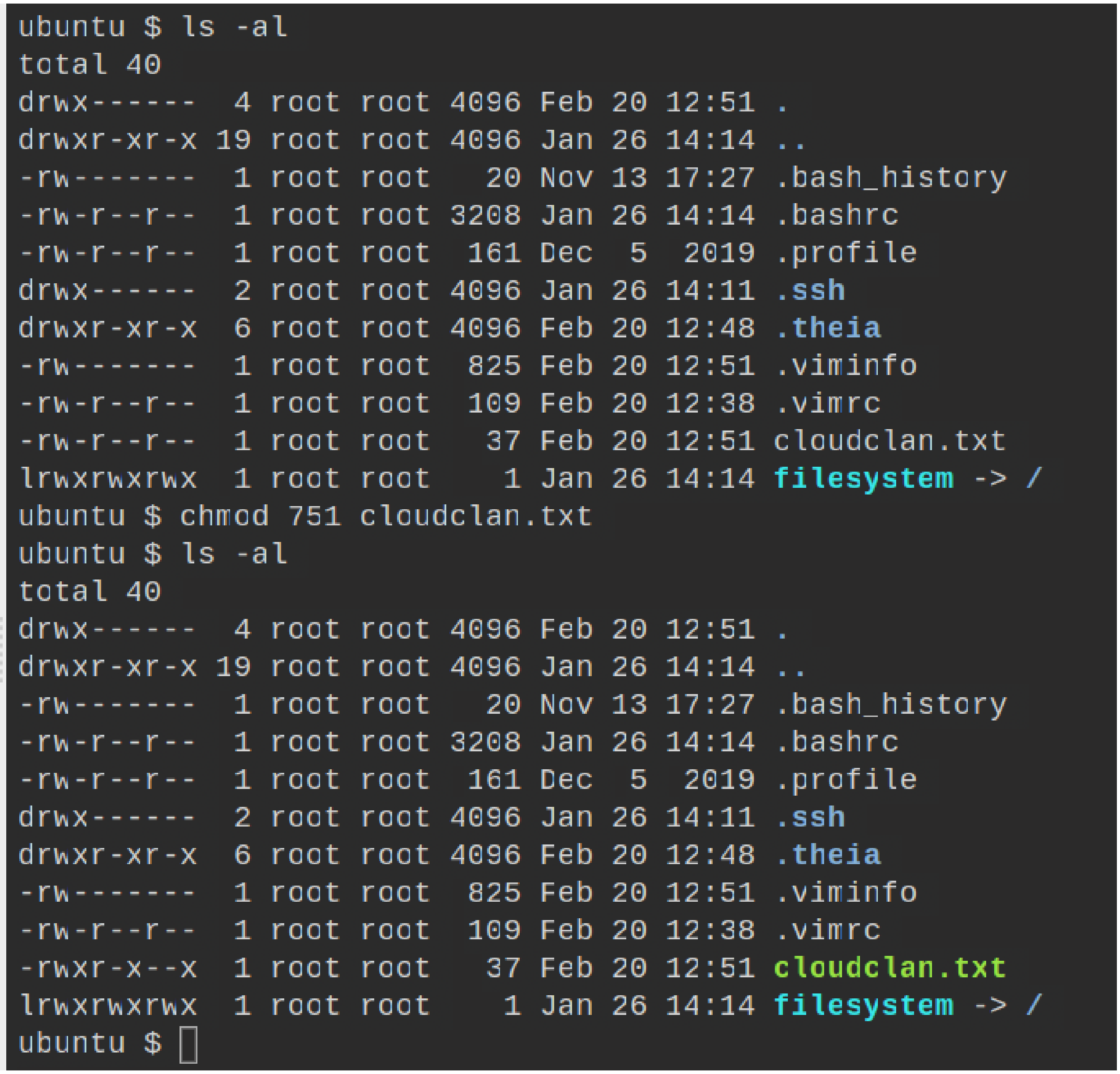
Link for reference: https://linuxize.com/post/linux-chown-command/

Using the **touch** command, create a file named **cloudclan.txt**.



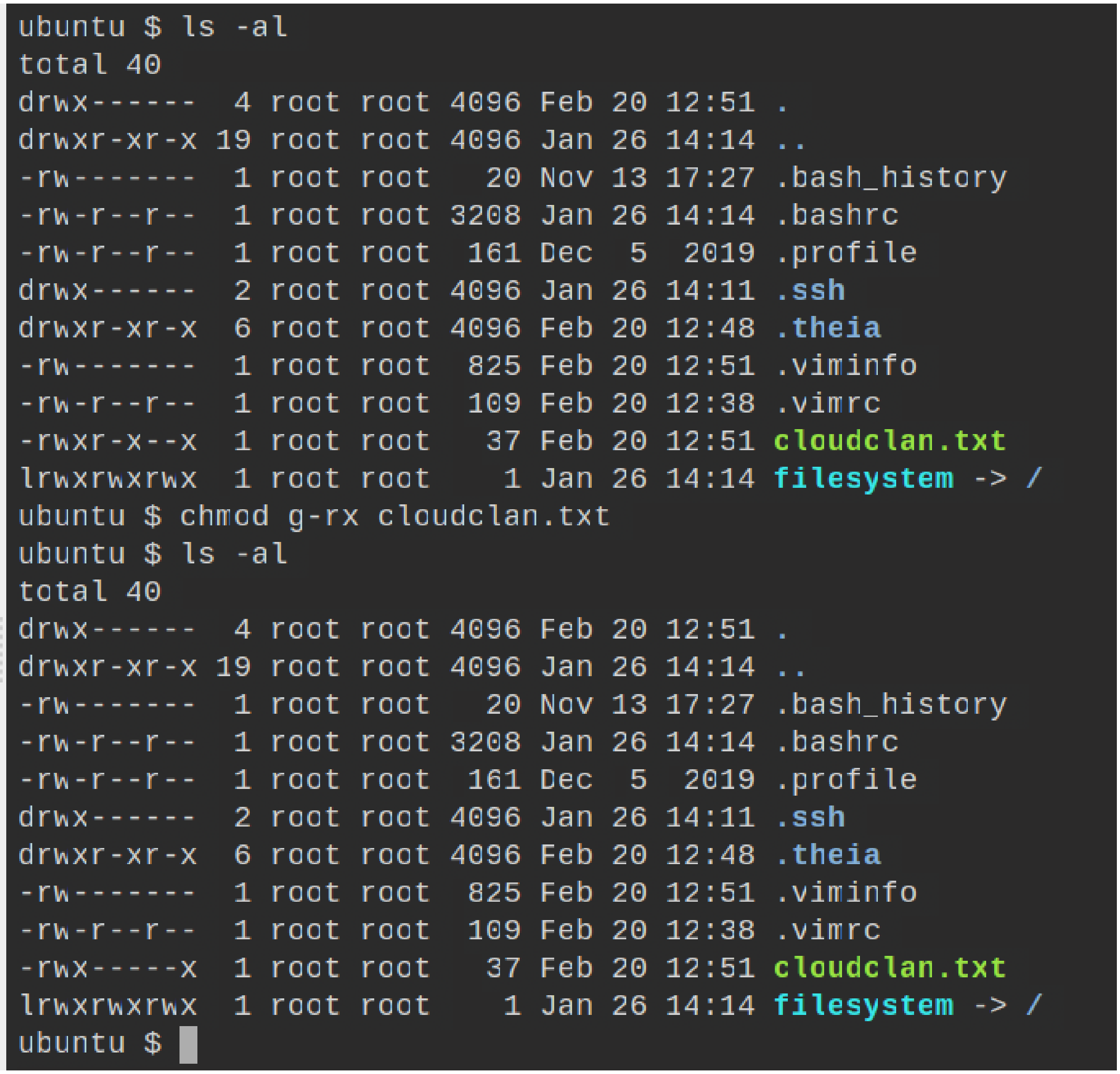
```
ubuntu $ su suyash
suyash@ubuntu:/root$ touch cloudclan.txt
touch: cannot touch 'cloudclan.txt': Permission denied
suyash@ubuntu:/root$ exit
exit
ubuntu $ touch cloudclan.txt
ubuntu $ vim cloudclan.txt
ubuntu $ cat cloudclan.txt

hello this is created by root user.
ubuntu $ 
```

Using the **chmod** command, change the permissions of **cloudclan.txt** to allow the owner to read, write, and execute, the group to read and execute, and others to only execute.

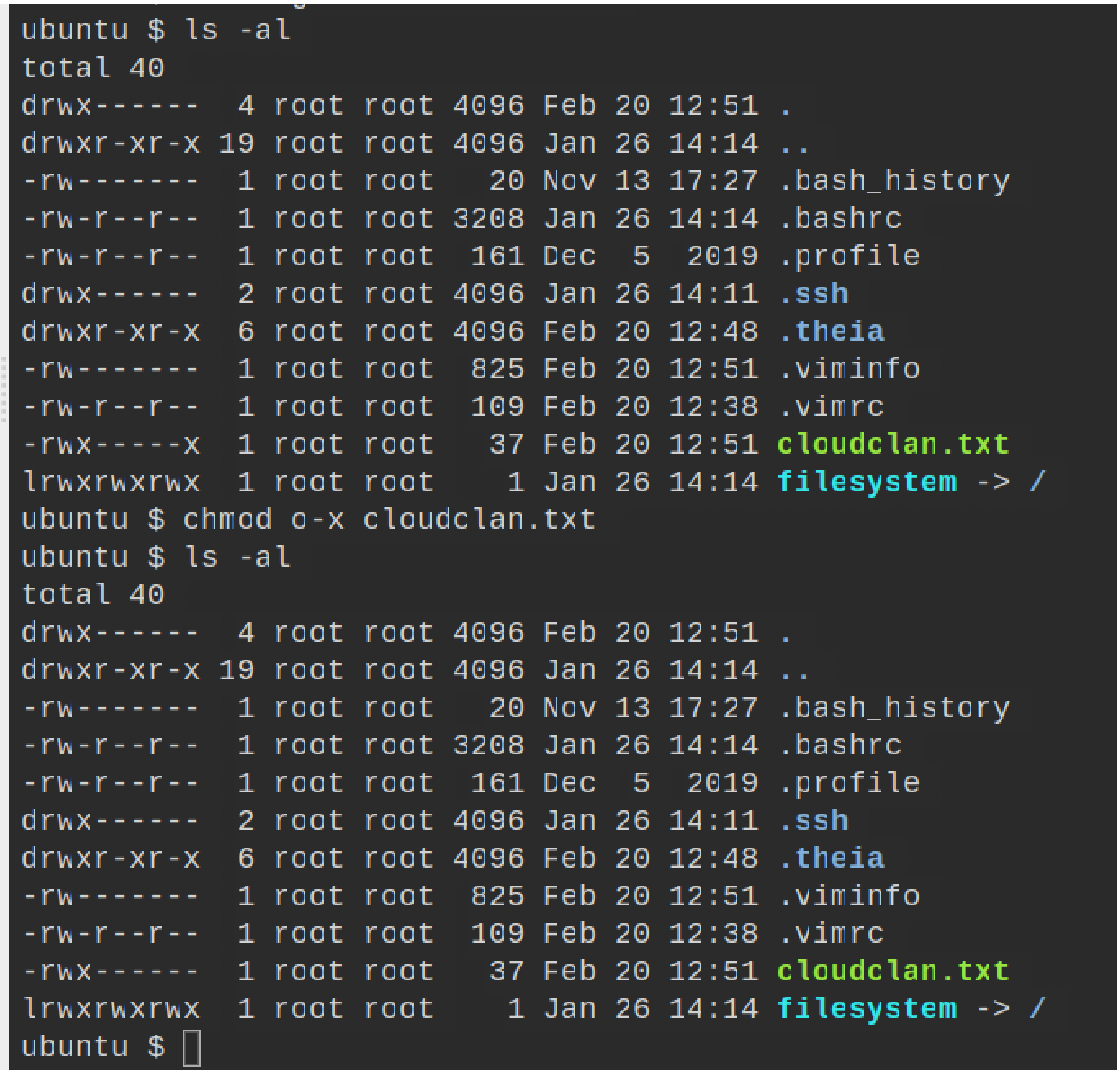


```
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rw-r--r--  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $ chmod 751 cloudclan.txt
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rwxr-x--x  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $
```

Using the **chmod** command, remove all permissions for the group and others on **cloudclan.txt**.



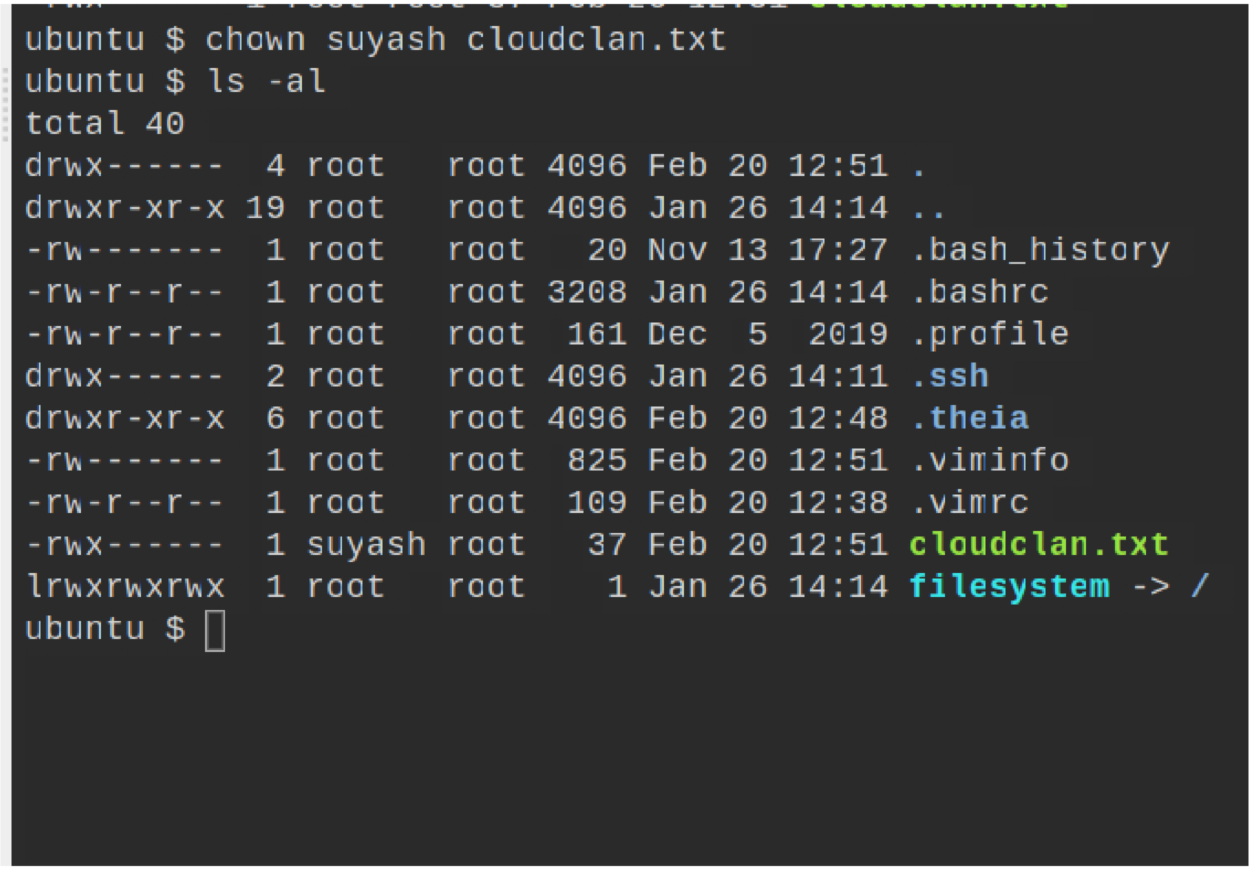
```
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rwxr-x--x  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $ chmod g-rx cloudclan.txt
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rwx-----x  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $ 
```



```
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rwx-----x  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $ chmod o-x cloudclan.txt
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rwx------  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $ 
```

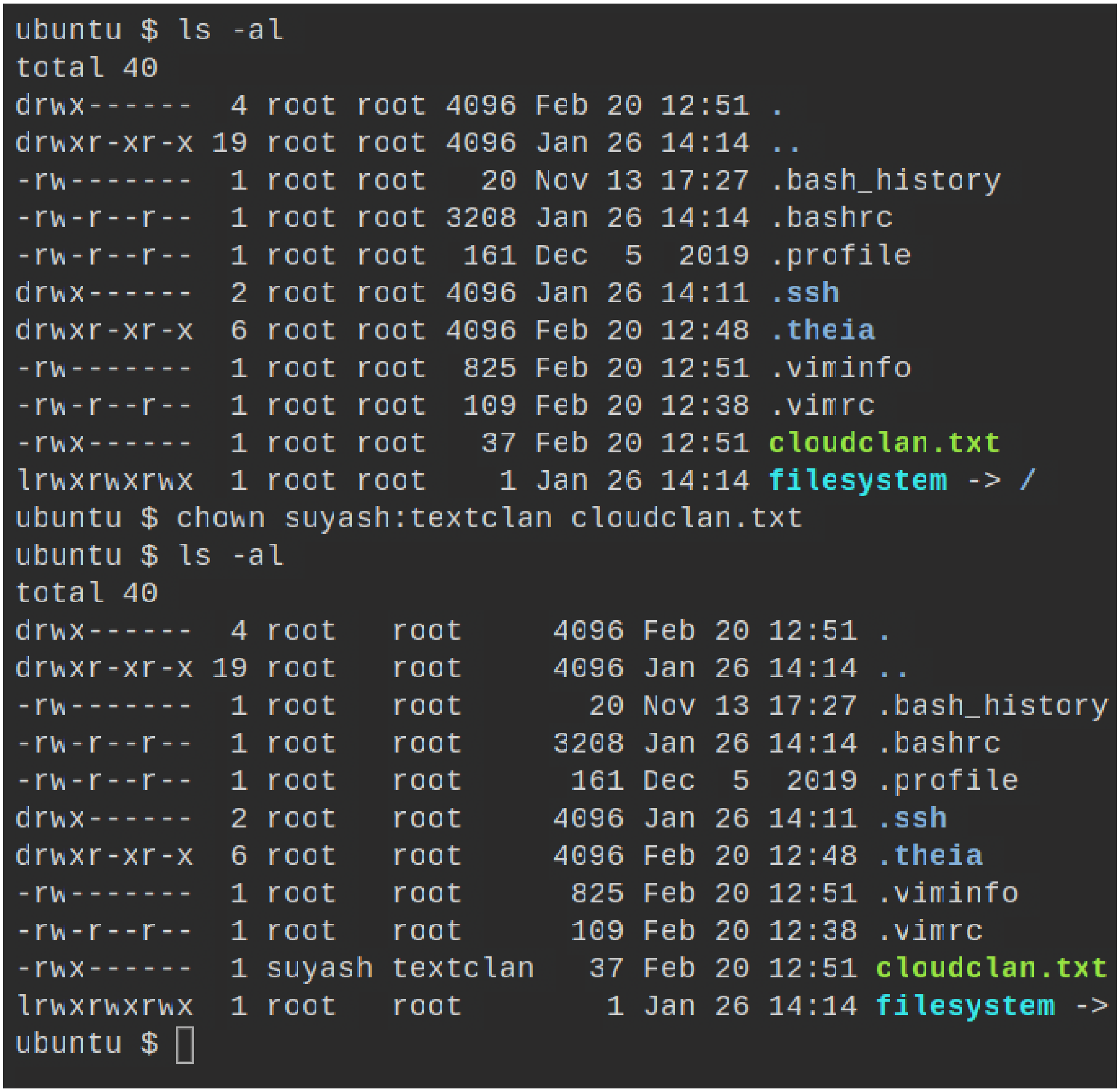
Using the **chmod** command, set the permissions of **cloudclan.txt** to allow the owner to read, write, and execute, the group to read and execute, and others to only execute.

Using the **chown** command, change the owner of **cloudclan.txt** to a specific user.



```
ubuntu $ chown suyash cloudclan.txt
ubuntu $ ls -al
total 40
drwx------  4 root   root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root   root 4096 Jan 26 14:14 ..
-rw-------  1 root   root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root   root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root   root  161 Dec  5  2019 .profile
drwx------  2 root   root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root   root 4096 Feb 20 12:48 .theia
-rw-------  1 root   root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root   root  109 Feb 20 12:38 .vimrc
-rwx------  1 suyash root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root   root    1 Jan 26 14:14 filesystem -> /
ubuntu $ 
```

Using the **chown** command, change the owner and group of **cloudclan.txt** to a specific user and group.



```
ubuntu $ ls -al
total 40
drwx------  4 root root 4096 Feb 20 12:51 .
drwxr-xr-x 19 root root 4096 Jan 26 14:14 ..
-rw-------  1 root root   20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root root 3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root root 4096 Feb 20 12:48 .theia
-rw-------  1 root root  825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root root  109 Feb 20 12:38 .vimrc
-rwx------  1 root root   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root root    1 Jan 26 14:14 filesystem -> /
ubuntu $ chown suyash:textclan cloudclan.txt
ubuntu $ ls -al
total 40
drwx------  4 root   root     4096 Feb 20 12:51 .
drwxr-xr-x 19 root   root     4096 Jan 26 14:14 ..
-rw-------  1 root   root       20 Nov 13 17:27 .bash_history
-rw-r--r--  1 root   root     3208 Jan 26 14:14 .bashrc
-rw-r--r--  1 root   root      161 Dec  5  2019 .profile
drwx------  2 root   root     4096 Jan 26 14:11 .ssh
drwxr-xr-x  6 root   root     4096 Feb 20 12:48 .theia
-rw-------  1 root   root      825 Feb 20 12:51 .viminfo
-rw-r--r--  1 root   root      109 Feb 20 12:38 .vimrc
-rwx------  1 suyash textclan   37 Feb 20 12:51 cloudclan.txt
lrwxrwxrwx  1 root   root        1 Jan 26 14:14 filesystem ->
ubuntu $
```

Extra:

Port:
In the context of server software, a port is a communication endpoint that identifies a specific process or service running on a computer. A port is identified by a number, typically ranging from 0 to 65535, and is used by networking protocols to route data to the appropriate application or service on the server.

In a server context, a port is typically associated with a specific network protocol, such as HTTP (port 80), HTTPS (port 443), or FTP (port 21). When a client requests data from a server using a specific protocol, the client typically connects to the server on the appropriate port, and the server software listens for incoming requests on that port.

For example, when you type a web address (URL) into a web browser, the browser sends an HTTP request to the web server on port 80 (or 443 for HTTPS). The web server then processes the request and sends the appropriate response back to the browser over the same port.

Different applications and services may use different ports, and in some cases, a server may run multiple services on different ports at the same time. However, not all ports are available for use by server applications, as some are reserved for specific purposes or restricted to certain users.