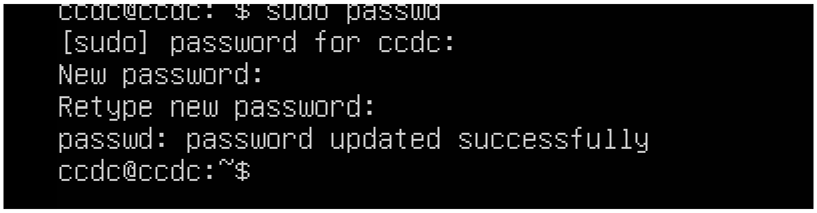


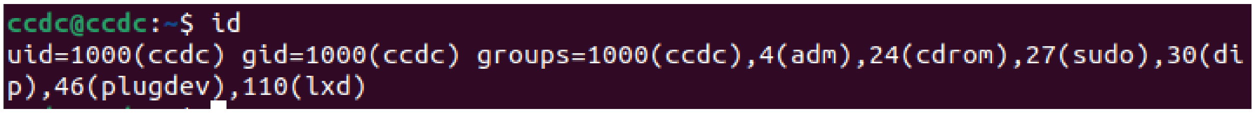
```
ccdc@ccdc:~$ sudo passwd
[sudo] password for ccdc:
New password:
Retype new password:
passwd: password updated successfully
ccdc@ccdc:~$
```

Problem: The vm provided logged me into tty. The problem is that I can't scroll the screen.

Hence, I did sudo apt-get install to install the GUI.

Problem: I tried to change the password but turns out I only changed the password for the root, not the user account. I found this explanation on stackoverflow about the difference, and did what they recommended there.The password was indeed modified when I rebooted the computer after that.

Users on this machine:

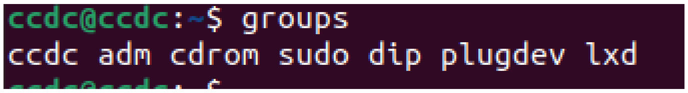


```
ccdc@ccdc:~$ id
uid=1000(ccdc) gid=1000(ccdc) groups=1000(ccdc),4(adm),24(cdrom),27(sudo),30(di
p),46(plugdev),110(lxd)
```

The output shows the ID and group ID of current user – ccdc

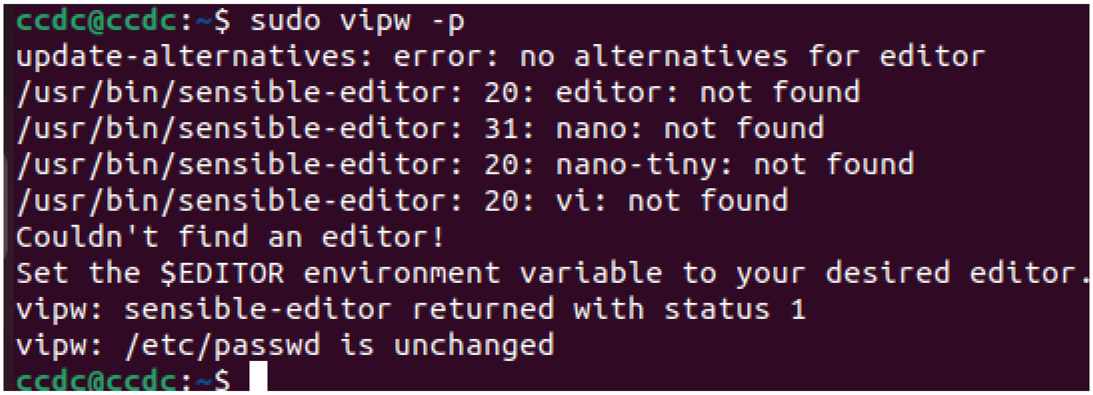
Problem: check user groups privilege

We have 7 groups:



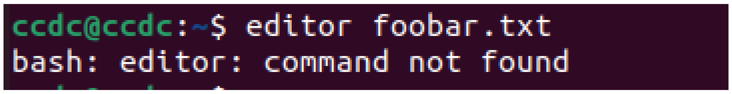
```
ccdc@ccdc:~$ groups
ccdc adm cdrom sudo dip plugdev lxd
ccdc@ccdc:~$
```

Problem: no editor, have to set up env variable for editor

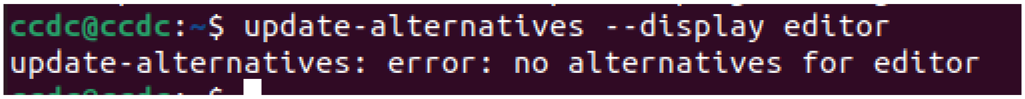


```
ccdc@ccdc:~$ sudo vipw -p
update-alternatives: error: no alternatives for editor
/usr/bin/sensible-editor: 20: editor: not found
/usr/bin/sensible-editor: 31: nano: not found
/usr/bin/sensible-editor: 20: nano-tiny: not found
/usr/bin/sensible-editor: 20: vi: not found
Couldn't find an editor!
Set the $EDITOR environment variable to your desired editor.
vipw: sensible-editor returned with status 1
vipw: /etc/passwd is unchanged
ccdc@ccdc:~$
```

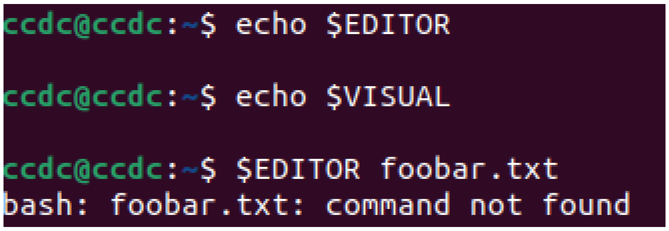
Tried to find default editor:



```
ccdc@ccdc:~$ editor foobar.txt
bash: editor: command not found
```



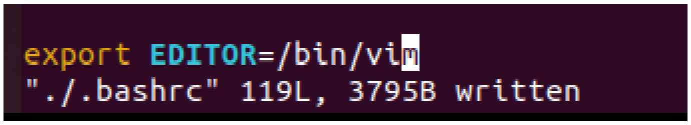
```
ccdc@ccdc:~$ update-alternatives --display editor
update-alternatives: error: no alternatives for editor
```



```
ccdc@ccdc:~$ echo $EDITOR

ccdc@ccdc:~$ echo $VISUAL

ccdc@ccdc:~$ $EDITOR foobar.txt
bash: foobar.txt: command not found
```

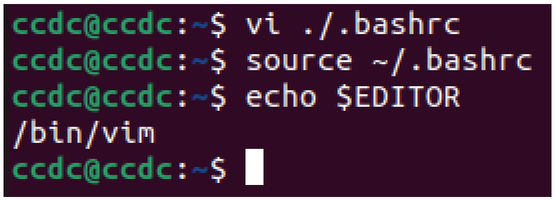
Problem: to add an editor, I need to access and modify ~/.bashrc. But now it doesn't have a default editor. What to do?

FIX: install an editor
Sudo apt install vim



```
export EDITOR=/bin/vim
"./.bashrc" 119L, 3795B written
```

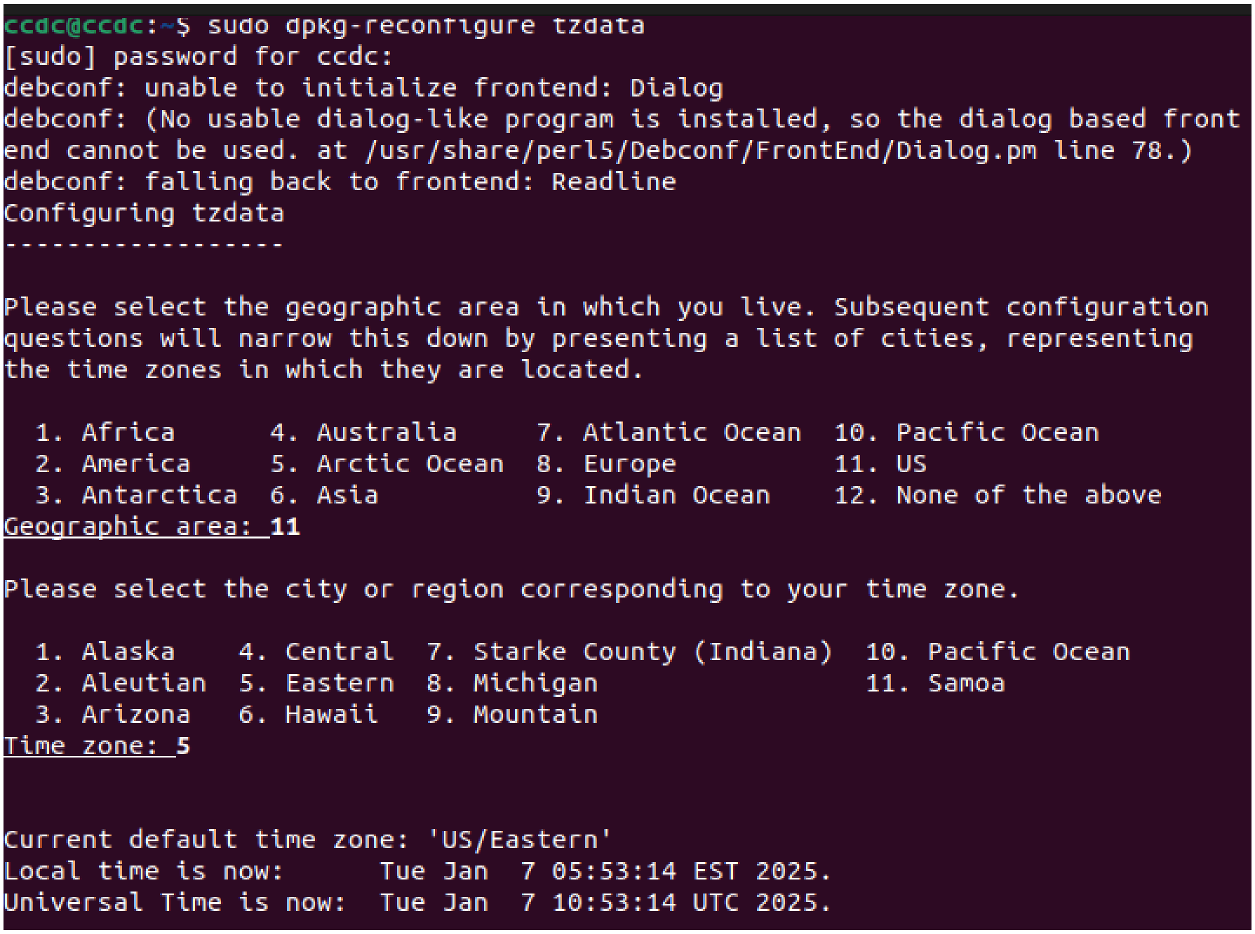
Added to ~/.bashrc



```
ccdc@ccdc:~$ vi ./.bashrc
ccdc@ccdc:~$ source ~/.bashrc
ccdc@ccdc:~$ echo $EDITOR
/bin/vim
ccdc@ccdc:~$ 
```

Then run source ~/.bashrc. The editor is now set to vim.

Problem: incorrect time display



```
ccdc@ccdc:~$ sudo dpkg-reconfigure tzdata
[sudo] password for ccdc:
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based front
end cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 78.)
debconf: falling back to frontend: Readline
Configuring tzdata
------------------

Please select the geographic area in which you live. Subsequent configuration
questions will narrow this down by presenting a list of cities, representing
the time zones in which they are located.

  1. Africa      4. Australia     7. Atlantic Ocean  10. Pacific Ocean
  2. America     5. Arctic Ocean  8. Europe          11. US
  3. Antarctica  6. Asia          9. Indian Ocean    12. None of the above
Geographic area: 11

Please select the city or region corresponding to your time zone.

  1. Alaska    4. Central  7. Starke County (Indiana)  10. Pacific Ocean
  2. Aleutian  5. Eastern  8. Michigan                 11. Samoa
  3. Arizona   6. Hawaii   9. Mountain
Time zone: 5


Current default time zone: 'US/Eastern'
Local time is now:      Tue Jan  7 05:53:14 EST 2025.
Universal Time is now:  Tue Jan  7 10:53:14 UTC 2025.
```

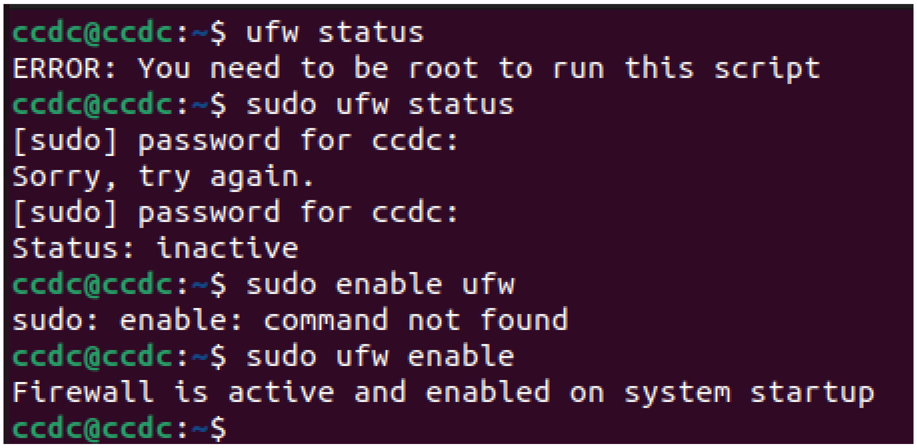
The time zone is correct by default. Using this command:

```
Unset
sudo dpkg-reconfigure tzdata
```

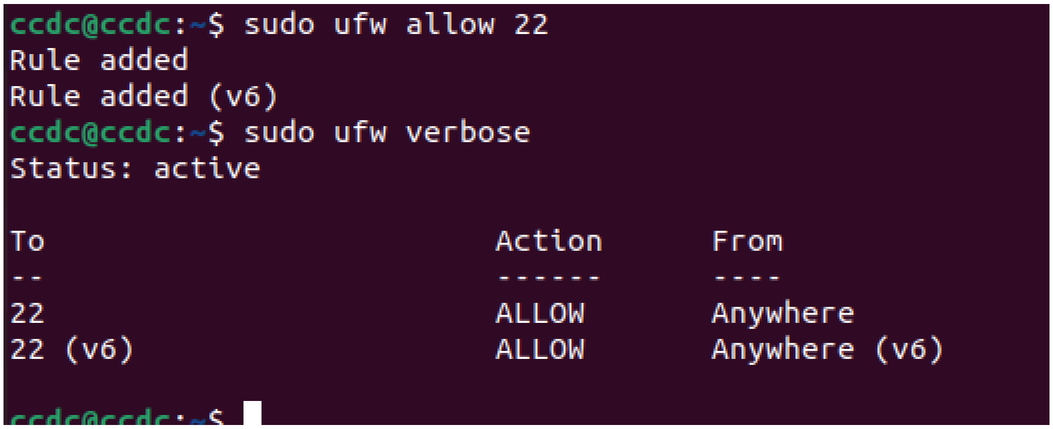
We can modify the time zone.

Problem:

Activating ufw by installing and enabling ufw as root



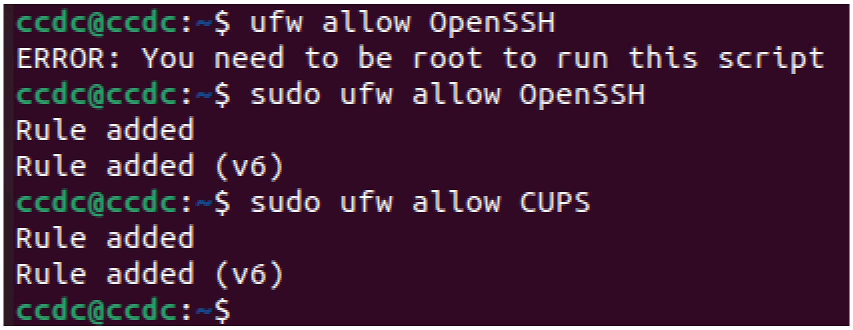
```
ccdc@ccdc:~$ ufw status
ERROR: You need to be root to run this script
ccdc@ccdc:~$ sudo ufw status
[sudo] password for ccdc:
Sorry, try again.
[sudo] password for ccdc:
Status: inactive
ccdc@ccdc:~$ sudo enable ufw
sudo: enable: command not found
ccdc@ccdc:~$ sudo ufw enable
Firewall is active and enabled on system startup
ccdc@ccdc:~$
```



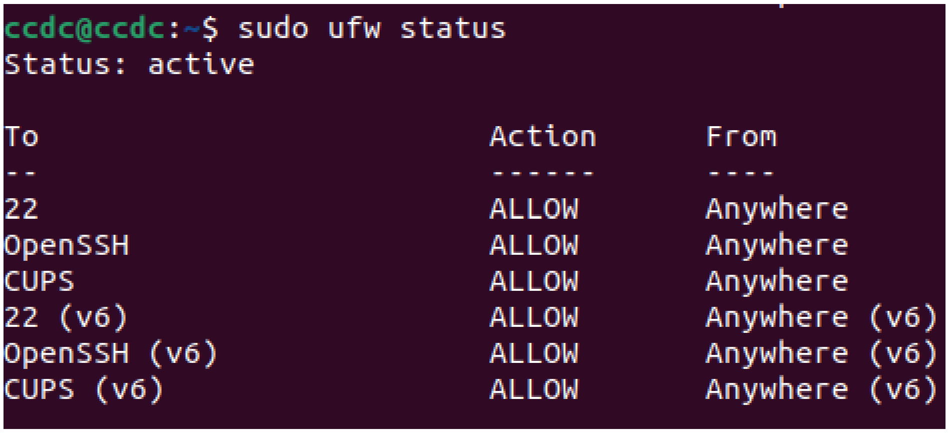
```
ccdc@ccdc:~$ sudo ufw allow 22
Rule added
Rule added (v6)
ccdc@ccdc:~$ sudo ufw verbose
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)

ccdc@ccdc:~$
```

The result above means that the firewall allows SSH access via port 22 from anywhere on the internet.
We need to enable the OpenSSH UFW application to make sure that the SSH port is open to connections so that we are able to log into our server remotely.

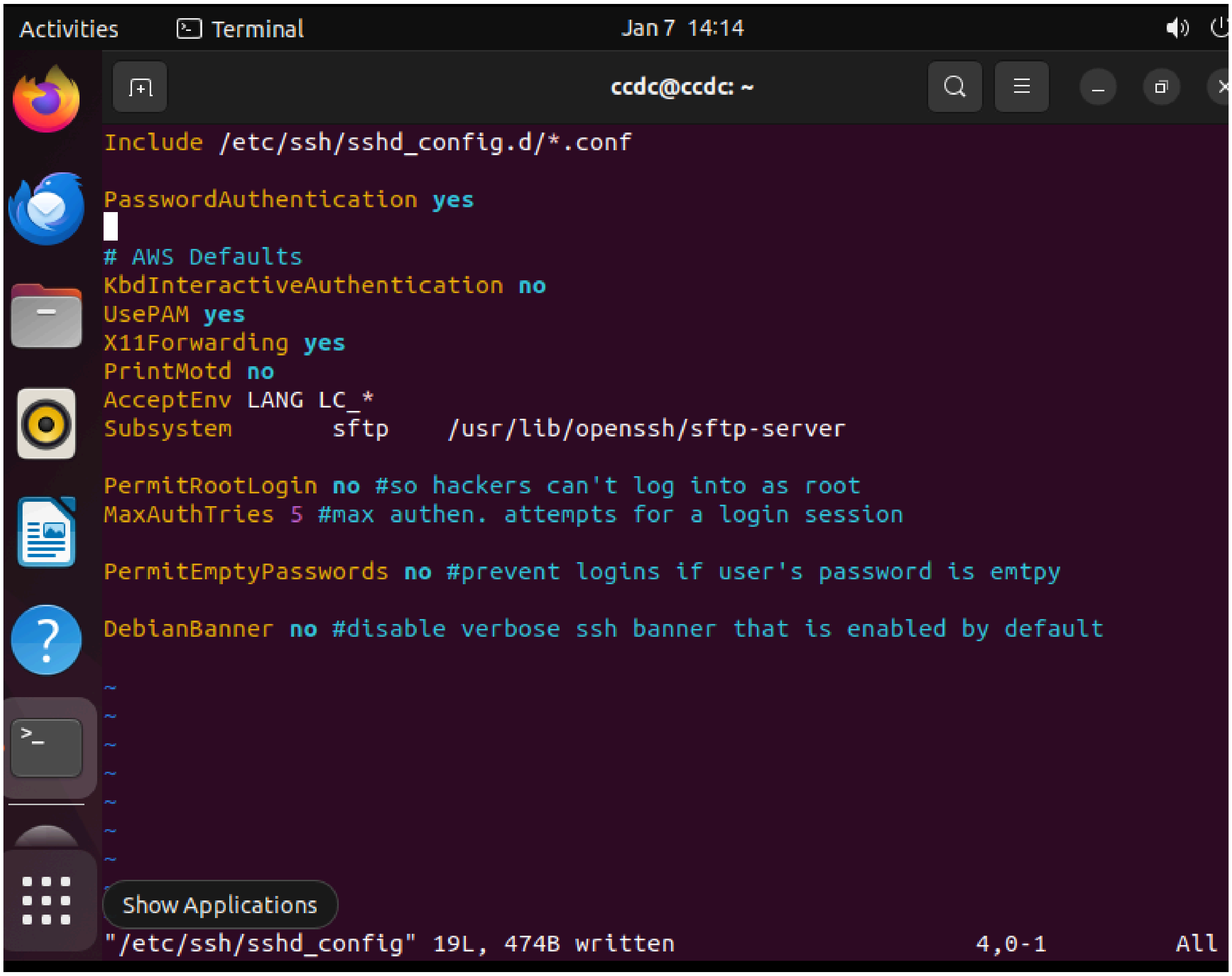
Problem: set up a basic firewall to make sure only connections to certain services are allowed.
What shows up means that these applications have profiles registered with ufw



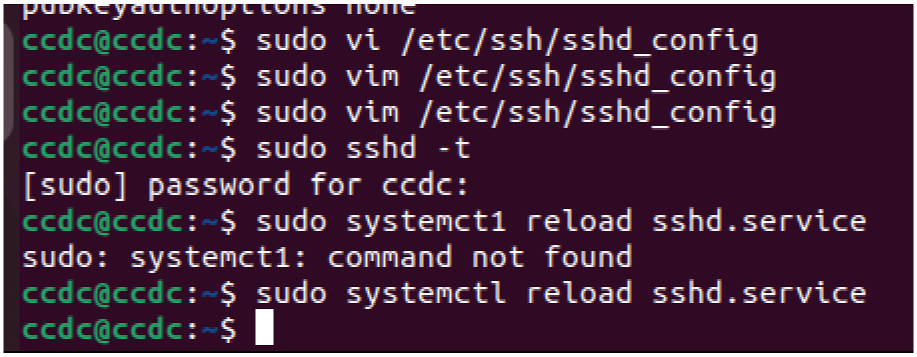
```
ccdc@ccdc:~$ ufw allow OpenSSH
ERROR: You need to be root to run this script
ccdc@ccdc:~$ sudo ufw allow OpenSSH
Rule added
Rule added (v6)
ccdc@ccdc:~$ sudo ufw allow CUPS
Rule added
Rule added (v6)
ccdc@ccdc:~$
```



```
ccdc@ccdc:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
CUPS                       ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
CUPS (v6)                  ALLOW       Anywhere (v6)
```



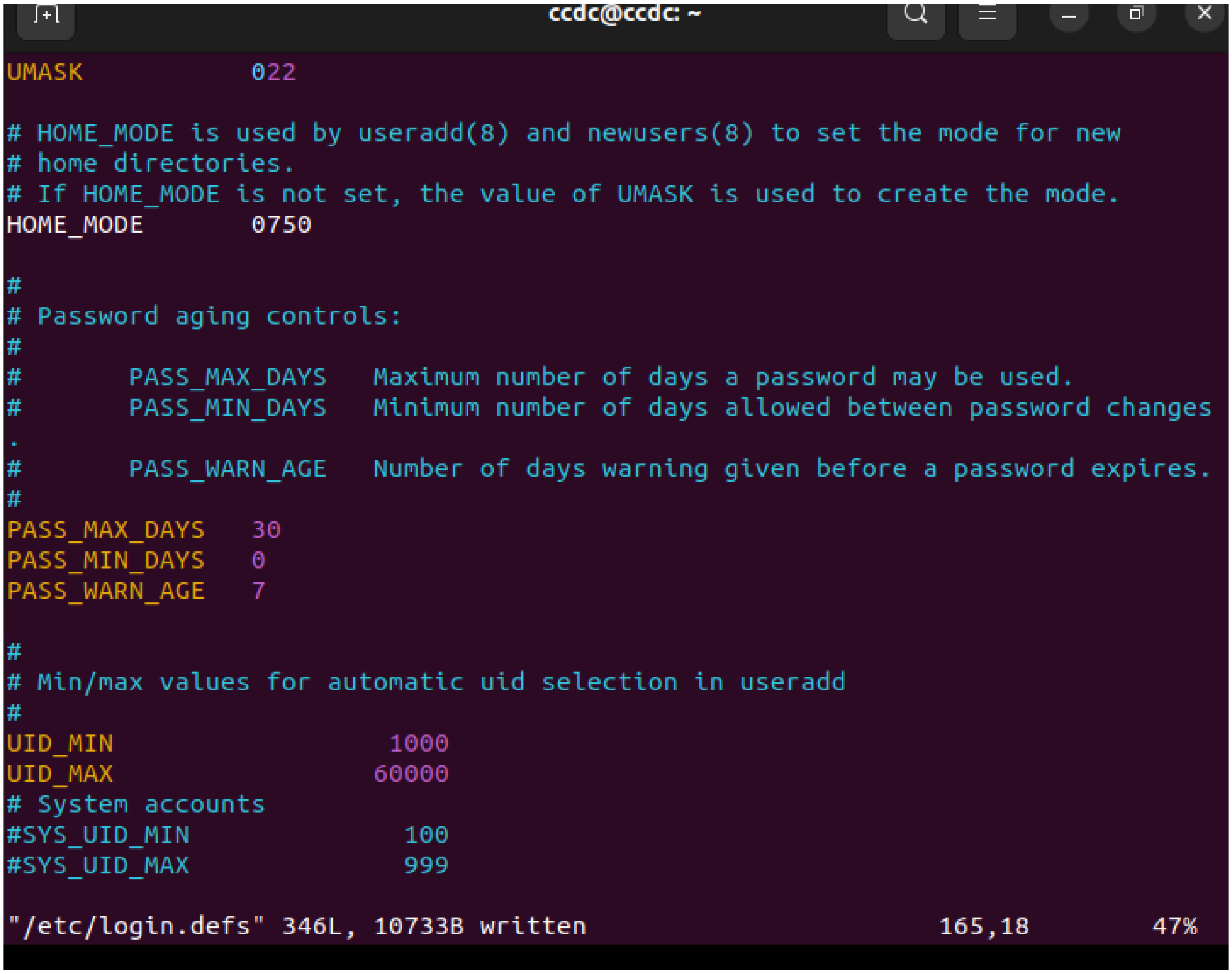
```
Include /etc/ssh/sshd_config.d/*.conf

PasswordAuthentication yes

# AWS Defaults
KbdInteractiveAuthentication no
UsePAM yes
X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
Subsystem       sftp    /usr/lib/openssh/sftp-server

PermitRootLogin no #so hackers can't log into as root
MaxAuthTries 5 #max authen. attempts for a login session

PermitEmptyPasswords no #prevent logins if user's password is emtpy

DebianBanner no #disable verbose ssh banner that is enabled by default

~
~
~
~
~
~
~
~
"/etc/ssh/sshd_config" 19L, 474B written                    4,0-1         All
```



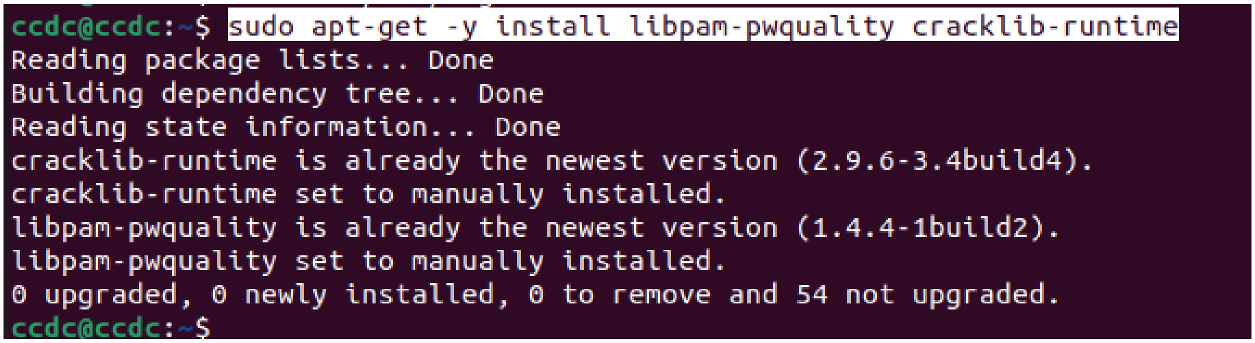
```
pubkeyauthoptions none
ccdc@ccdc:~$ sudo vi /etc/ssh/sshd_config
ccdc@ccdc:~$ sudo vim /etc/ssh/sshd_config
ccdc@ccdc:~$ sudo vim /etc/ssh/sshd_config
ccdc@ccdc:~$ sudo sshd -t
[sudo] password for ccdc:
ccdc@ccdc:~$ sudo systemct1 reload sshd.service
sudo: systemct1: command not found
ccdc@ccdc:~$ sudo systemctl reload sshd.service
ccdc@ccdc:~$ 
```

<u>Changing password policies</u>

sudo vi /etc/login.defs → change PASS_MAX_DAYS from 99999 to 30
Enforcing all users to change passwords every 30 days or less.
Next, to set the default password requirements, we use the pwquality/pam_pwqulaity PAM module: first install lib-am-pwquality package

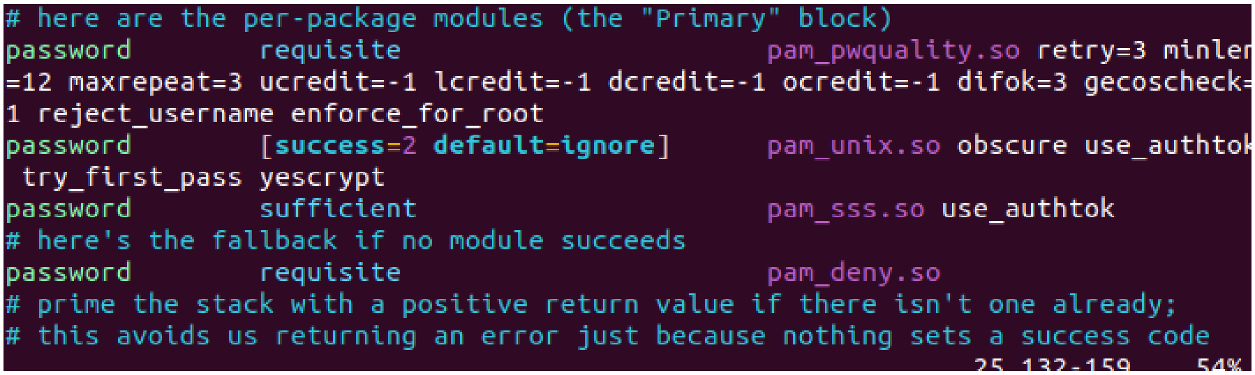


```
UMASK           022

# HOME_MODE is used by useradd(8) and newusers(8) to set the mode for new
# home directories.
# If HOME_MODE is not set, the value of UMASK is used to create the mode.
HOME_MODE       0750

#
# Password aging controls:
#
#       PASS_MAX_DAYS   Maximum number of days a password may be used.
#       PASS_MIN_DAYS   Minimum number of days allowed between password changes
.
#       PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   30
PASS_MIN_DAYS   0
PASS_WARN_AGE   7

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN                  1000
UID_MAX                 60000
# System accounts
#SYS_UID_MIN              100
#SYS_UID_MAX              999

"/etc/login.defs" 346L, 10733B written                        165,18        47%
```



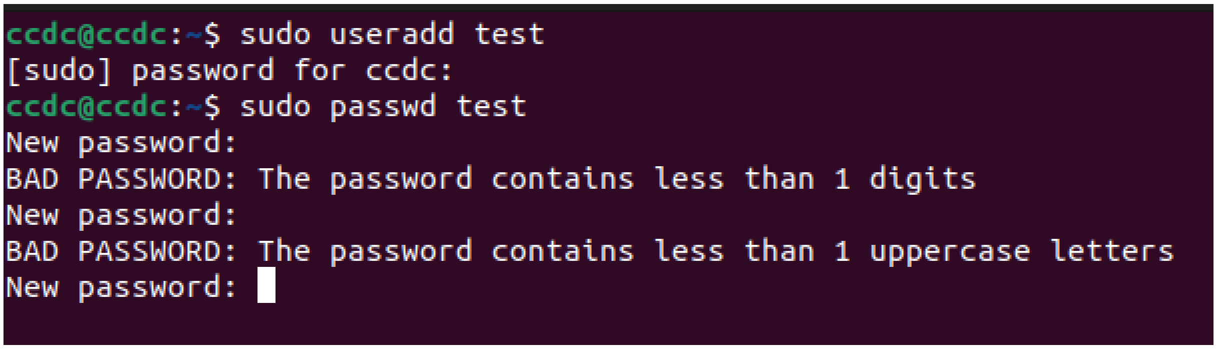
```
ccdc@ccdc:~$ sudo apt-get -y install libpam-pwquality cracklib-runtime
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cracklib-runtime is already the newest version (2.9.6-3.4build4).
cracklib-runtime set to manually installed.
libpam-pwquality is already the newest version (1.4.4-1build2).
libpam-pwquality set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 54 not upgraded.
ccdc@ccdc:~$
```

In sudo /etc/pam.d/common-password, find line 25



```
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_pwquality.so retry=3 minlen
=12 maxrepeat=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 difok=3 gecoscheck=
1 reject_username enforce_for_root
password        [success=2 default=ignore]      pam_unix.so obscure use_authtok
 try_first_pass yescrypt
password        sufficient                      pam_sss.so use_authtok
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
                                                                  25,132-159    54%
```

Then reboot our server with the command sudo reboot

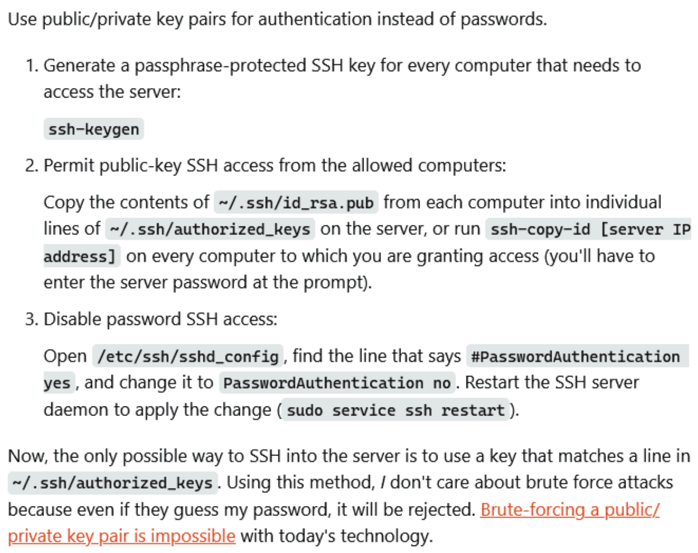


```
ccdc@ccdc:~$ sudo useradd test
[sudo] password for ccdc:
ccdc@ccdc:~$ sudo passwd test
New password:
BAD PASSWORD: The password contains less than 1 digits
New password:
BAD PASSWORD: The password contains less than 1 uppercase letters
New password:
```

The first password test I put in “test” and “testtesttest1” for the second one, which gives me the corresponding results above.

implementing strong password policies tutorial
another good tutorial that I haven't followed through

Problem: use public/private key pairs for SSH authentication instead of passwords



Use public/private key pairs for authentication instead of passwords.

1. Generate a passphrase-protected SSH key for every computer that needs to access the server:

   `ssh-keygen`

2. Permit public-key SSH access from the allowed computers:

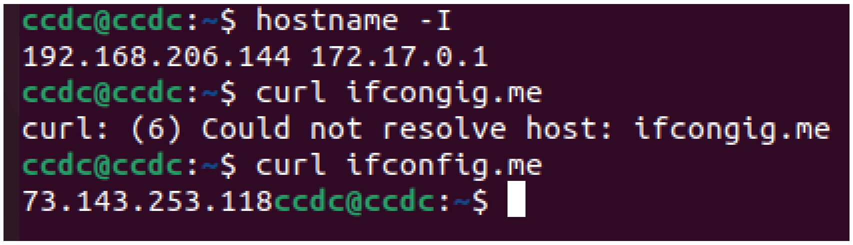
   Copy the contents of `~/.ssh/id_rsa.pub` from each computer into individual lines of `~/.ssh/authorized_keys` on the server, or run `ssh-copy-id [server IP address]` on every computer to which you are granting access (you'll have to enter the server password at the prompt).

3. Disable password SSH access:

   Open `/etc/ssh/sshd_config`, find the line that says `#PasswordAuthentication yes`, and change it to `PasswordAuthentication no`. Restart the SSH server daemon to apply the change (`sudo service ssh restart`).

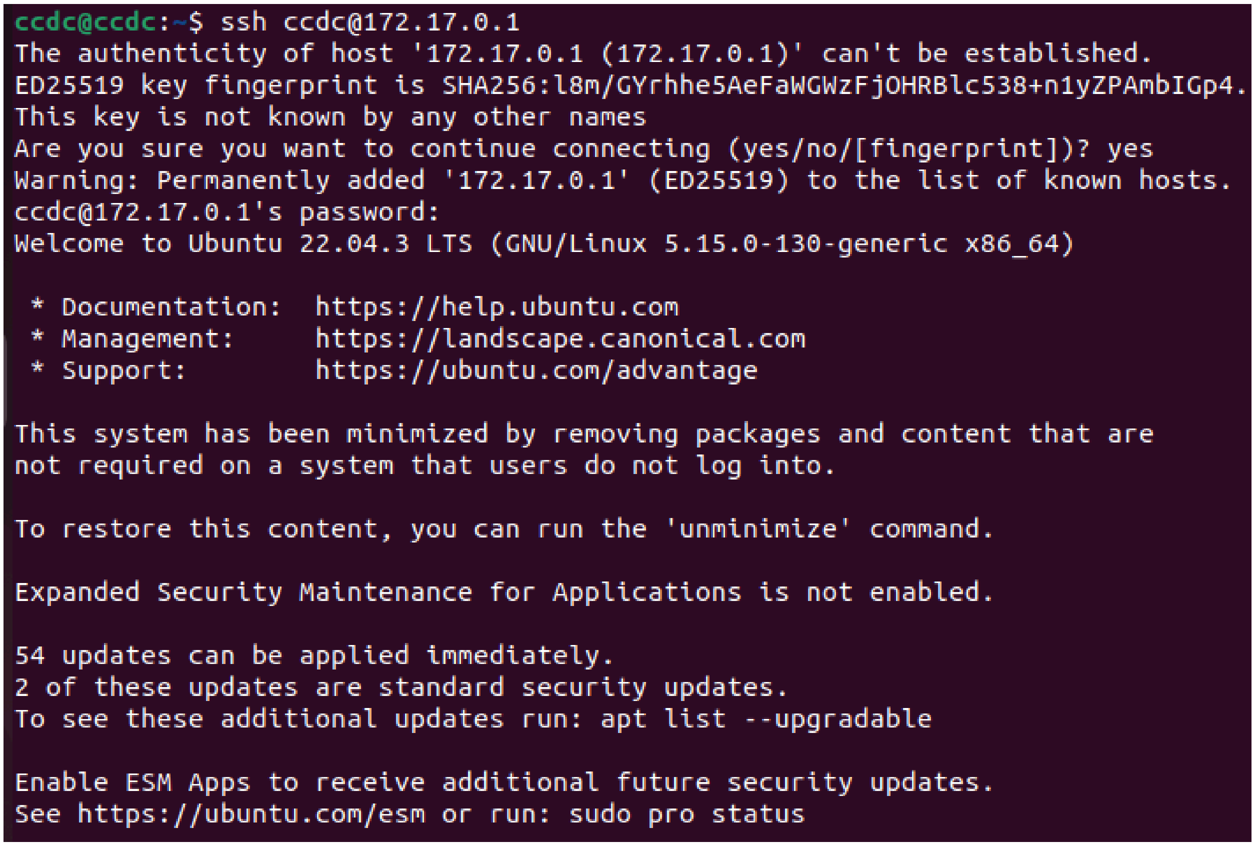
Now, the only possible way to SSH into the server is to use a key that matches a line in `~/.ssh/authorized_keys`. Using this method, *I* don't care about brute force attacks because even if they guess my password, it will be rejected. Brute-forcing a public/private key pair is impossible with today's technology.

I will follow this guide from stackoverflow.



```
ccdc@ccdc:~$ hostname -I
192.168.206.144 172.17.0.1
ccdc@ccdc:~$ curl ifcongig.me
curl: (6) Could not resolve host: ifcongig.me
ccdc@ccdc:~$ curl ifconfig.me
73.143.253.118ccdc@ccdc:~$
```

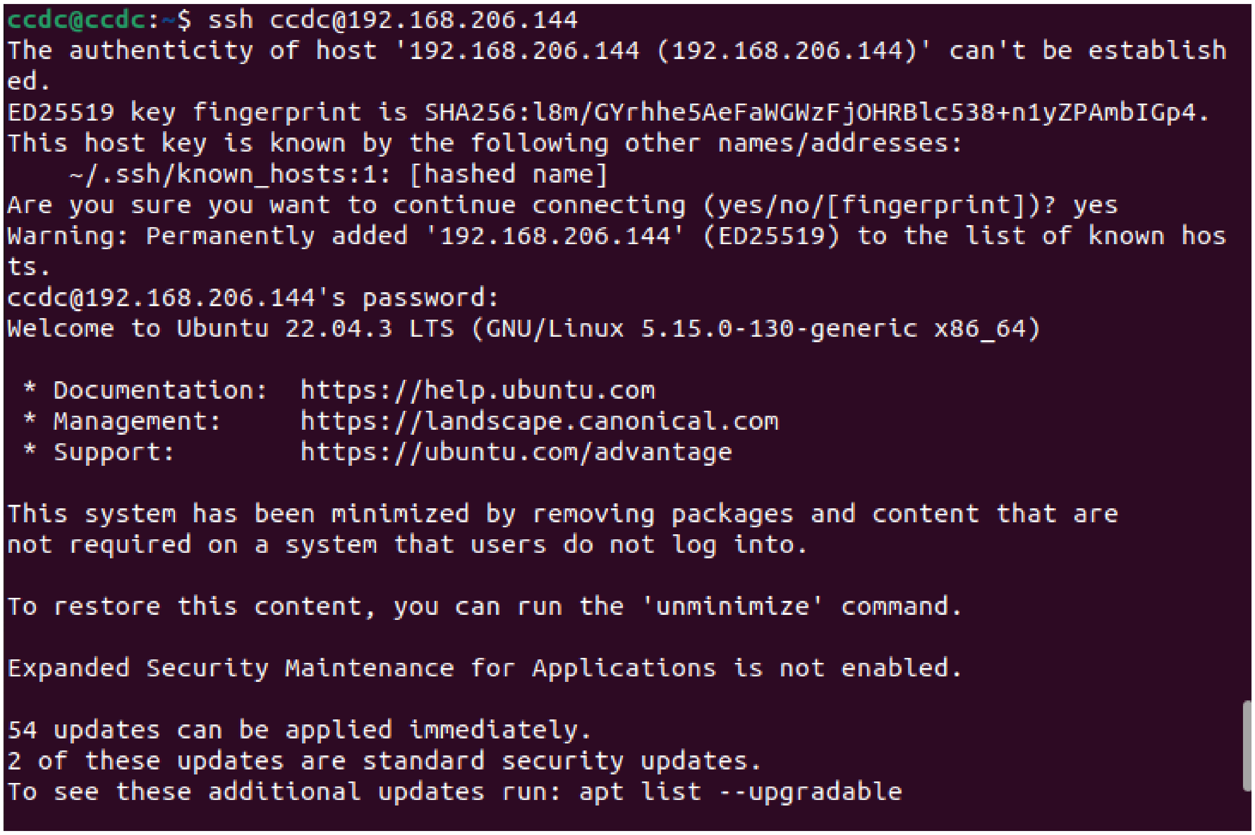
We got the private IP addresses from the command hostname -I.



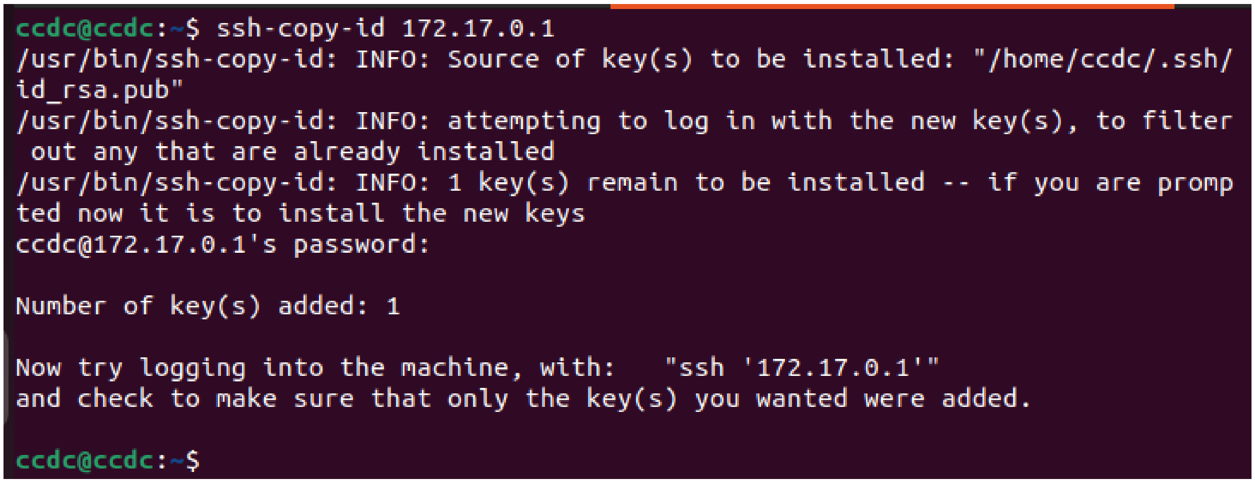
```
ccdc@ccdc:~$ ssh ccdc@172.17.0.1
The authenticity of host '172.17.0.1 (172.17.0.1)' can't be established.
ED25519 key fingerprint is SHA256:l8m/GYrhhe5AeFaWGWzFjOHRBlc538+n1yZPAmbIGp4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.17.0.1' (ED25519) to the list of known hosts.
ccdc@172.17.0.1's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-130-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
```

Added 172.17.0.1 – one of the private IP addresses– to the list of known hosts.

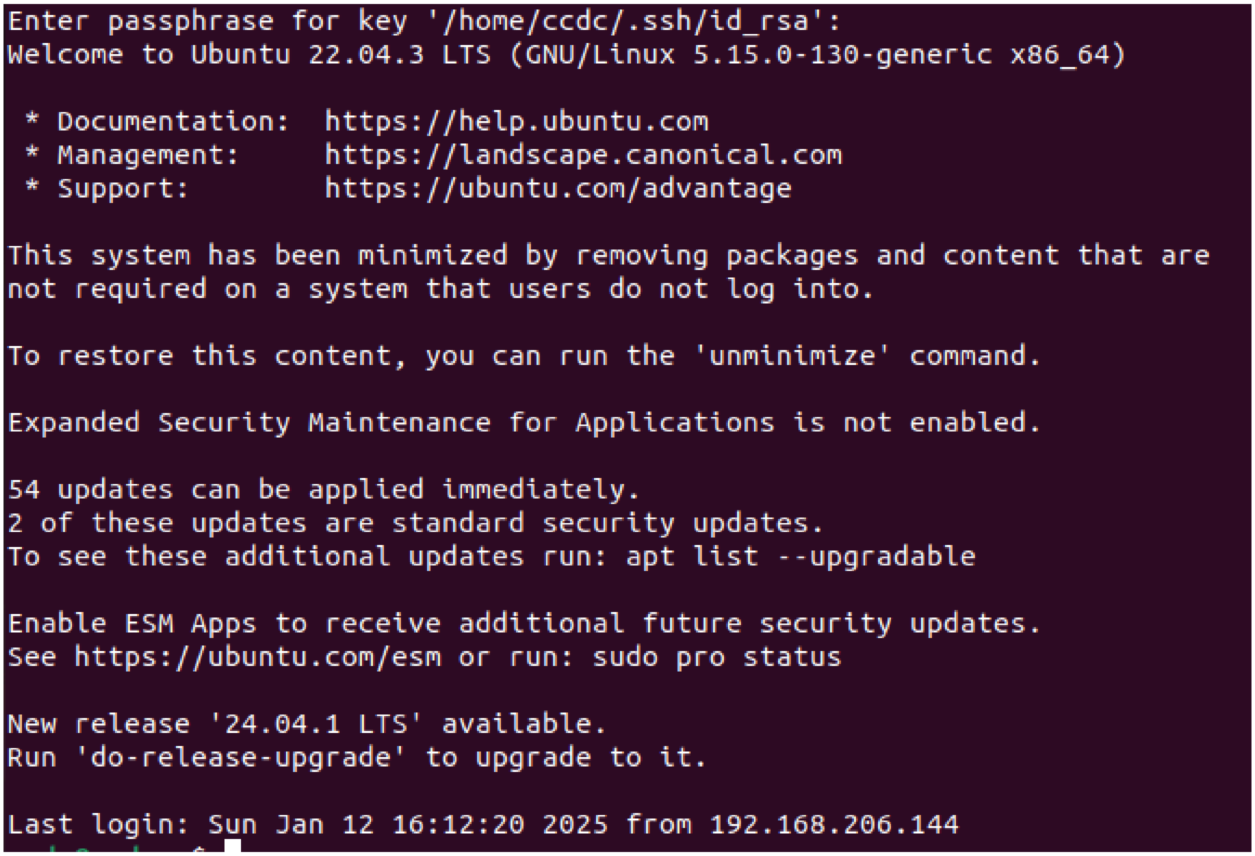
In a similar way, we can add the other private IP address to the list of known host:



```
ccdc@ccdc:~$ ssh ccdc@192.168.206.144
The authenticity of host '192.168.206.144 (192.168.206.144)' can't be establish
ed.
ED25519 key fingerprint is SHA256:l8m/GYrhhe5AeFaWGWzFjOHRBlc538+n1yZPAmbIGp4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.206.144' (ED25519) to the list of known hos
ts.
ccdc@192.168.206.144's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-130-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
```

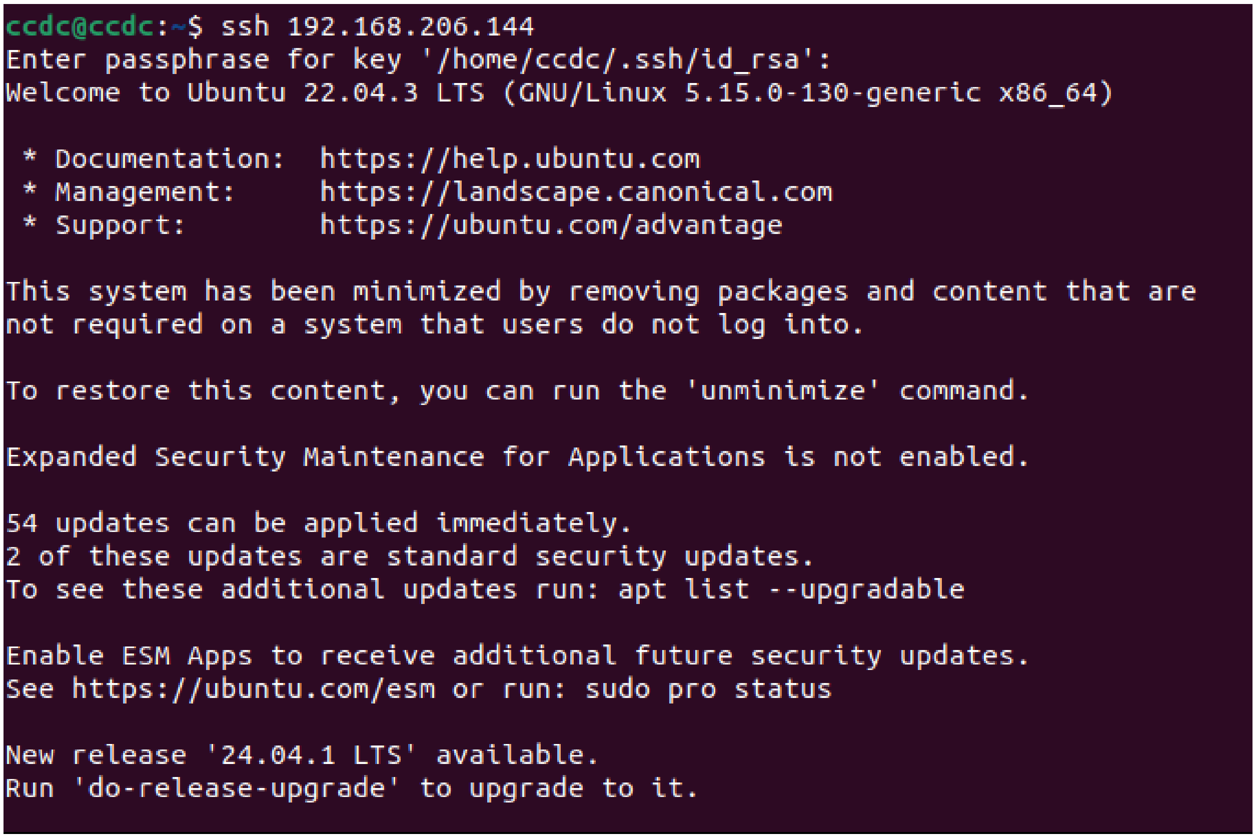


```
ccdc@ccdc:~$ ssh-copy-id 172.17.0.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ccdc/.ssh/
id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
 out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are promp
ted now it is to install the new keys
ccdc@172.17.0.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh '172.17.0.1'"
and check to make sure that only the key(s) you wanted were added.

ccdc@ccdc:~$
```



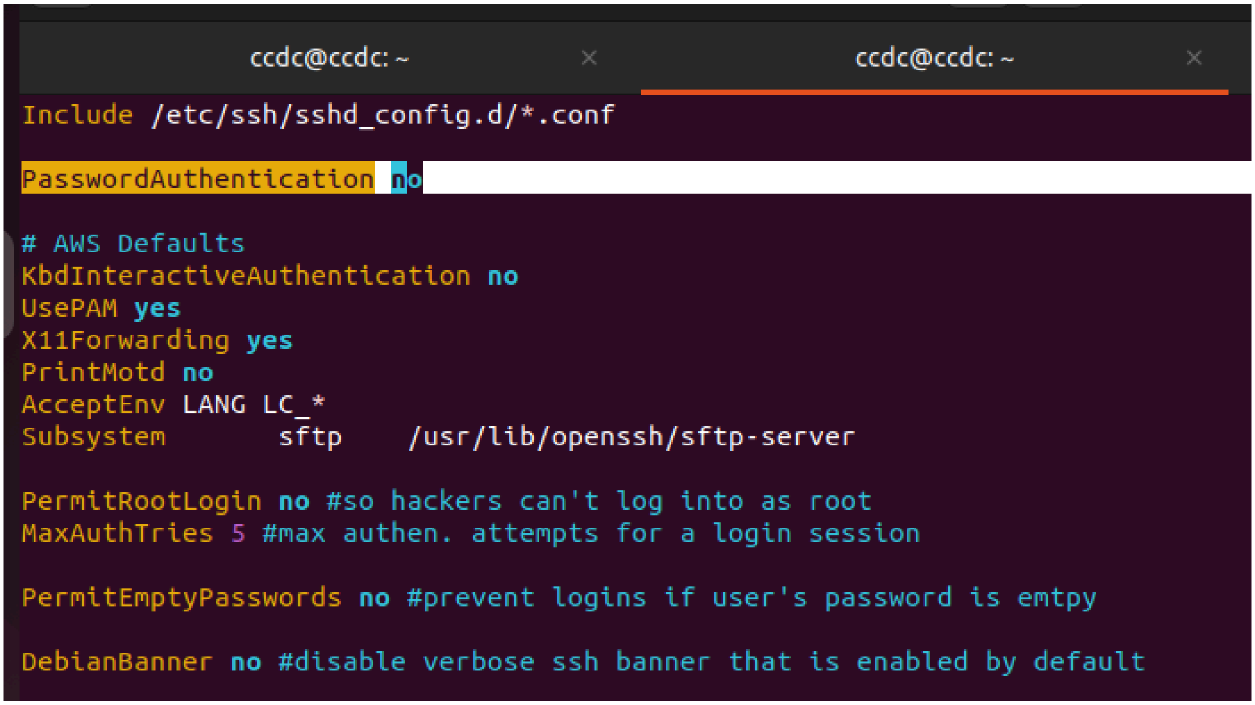
```
Enter passphrase for key '/home/ccdc/.ssh/id_rsa':
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-130-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Jan 12 16:12:20 2025 from 192.168.206.144
```

Now we can also SSH into the other IP address



```
ccdc@ccdc:~$ ssh 192.168.206.144
Enter passphrase for key '/home/ccdc/.ssh/id_rsa':
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-130-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

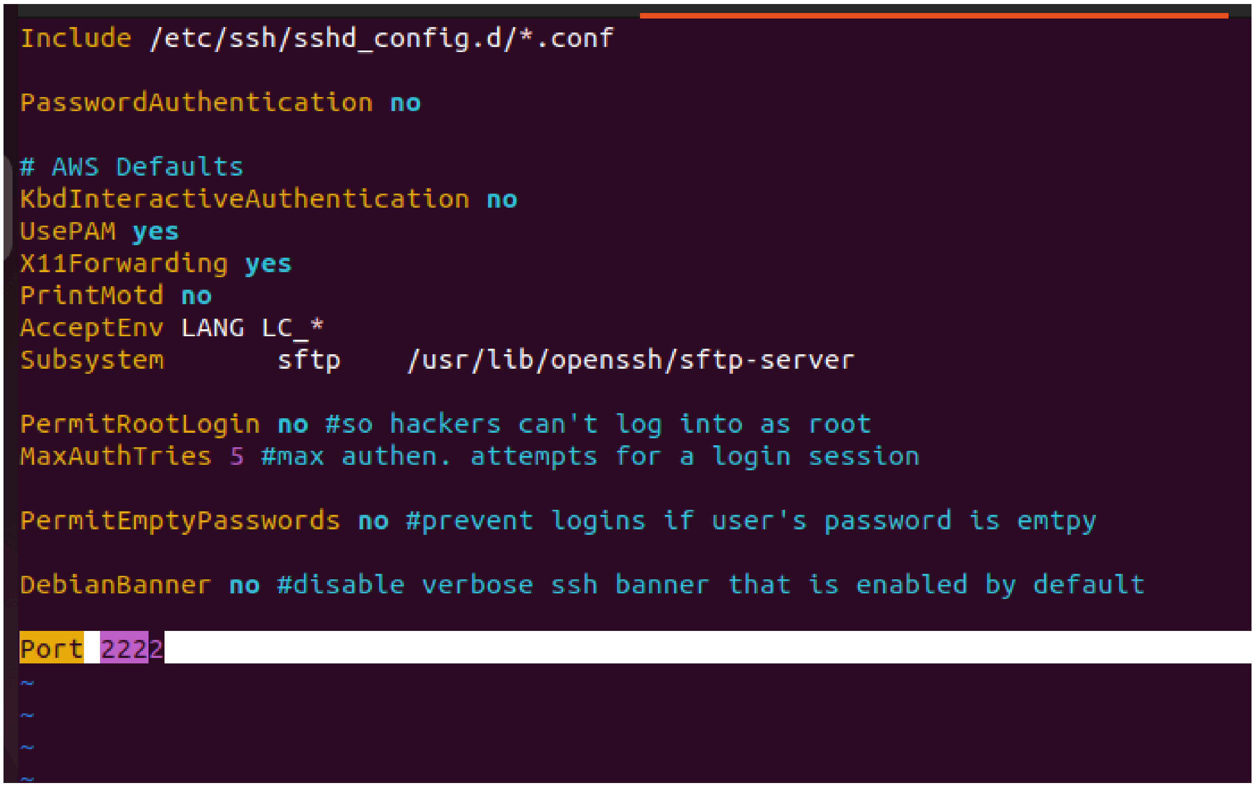
After we permit public-key SSH access, we need to disable password SSH access: in /etc/ssh/sshd_config, we change PasswordAuthentication from ‘yes’ to ‘no’



```
Include /etc/ssh/sshd_config.d/*.conf

PasswordAuthentication no

# AWS Defaults
KbdInteractiveAuthentication no
UsePAM yes
X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
Subsystem       sftp    /usr/lib/openssh/sftp-server

PermitRootLogin no #so hackers can't log into as root
MaxAuthTries 5 #max authen. attempts for a login session

PermitEmptyPasswords no #prevent logins if user's password is emtpy

DebianBanner no #disable verbose ssh banner that is enabled by default
```

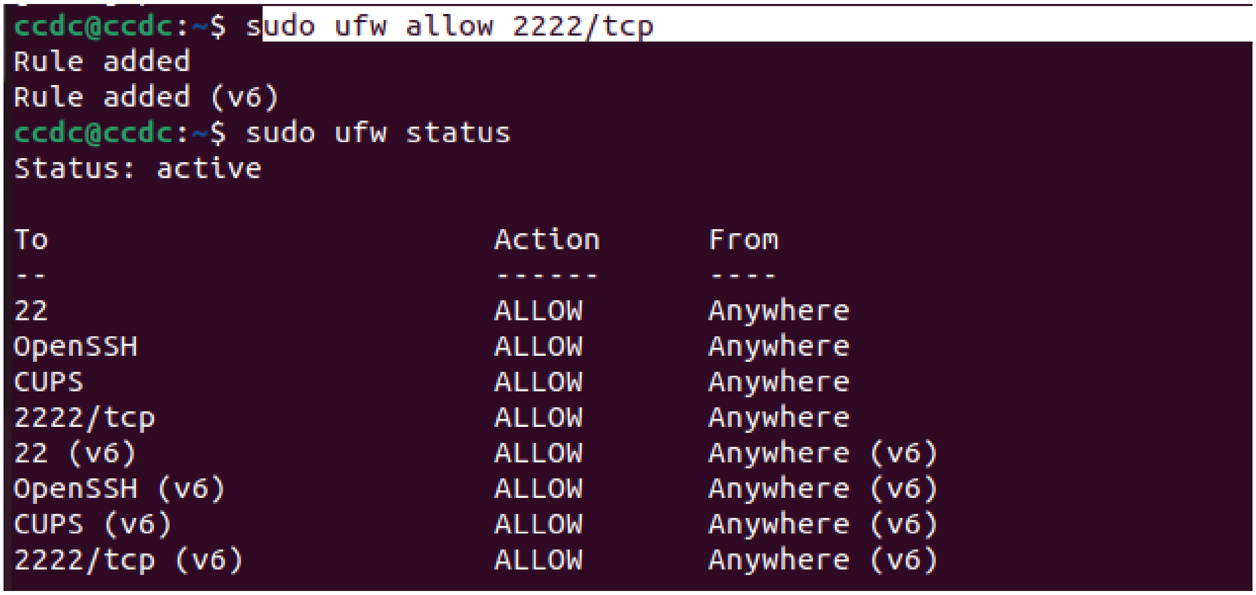
Then restart the SSH server daemon to apply the change with the command *sudo service ssh restart*
Although we need to enter a passphrase to use the private key to authenticate, it adds an extra layer of security.

Add port number 2222 into config file:



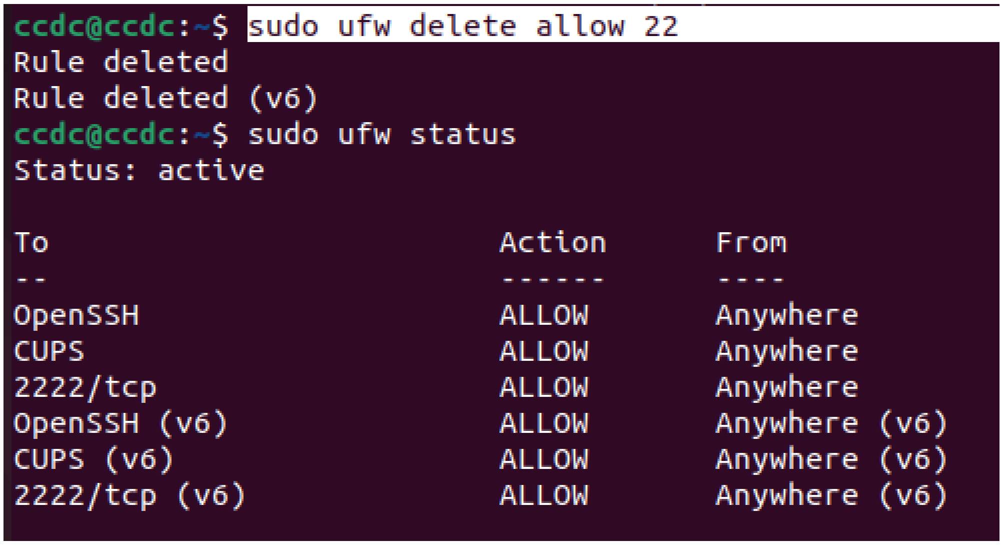
```
Include /etc/ssh/sshd_config.d/*.conf

PasswordAuthentication no

# AWS Defaults
KbdInteractiveAuthentication no
UsePAM yes
X11Forwarding yes
PrintMotd no
AcceptEnv LANG LC_*
Subsystem       sftp    /usr/lib/openssh/sftp-server

PermitRootLogin no #so hackers can't log into as root
MaxAuthTries 5 #max authen. attempts for a login session

PermitEmptyPasswords no #prevent logins if user's password is emtpy

DebianBanner no #disable verbose ssh banner that is enabled by default

Port 2222
~
~
~
~
```

The command below allows incoming TCP traffic on port 2222 by updating the UFW rules



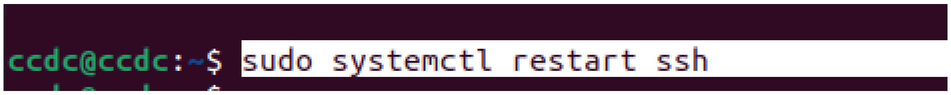
```
ccdc@ccdc:~$ sudo ufw allow 2222/tcp
Rule added
Rule added (v6)
ccdc@ccdc:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
CUPS                       ALLOW       Anywhere
2222/tcp                   ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)
CUPS (v6)                  ALLOW       Anywhere (v6)
2222/tcp (v6)              ALLOW       Anywhere (v6)
```

Then use the following command to remove port 22



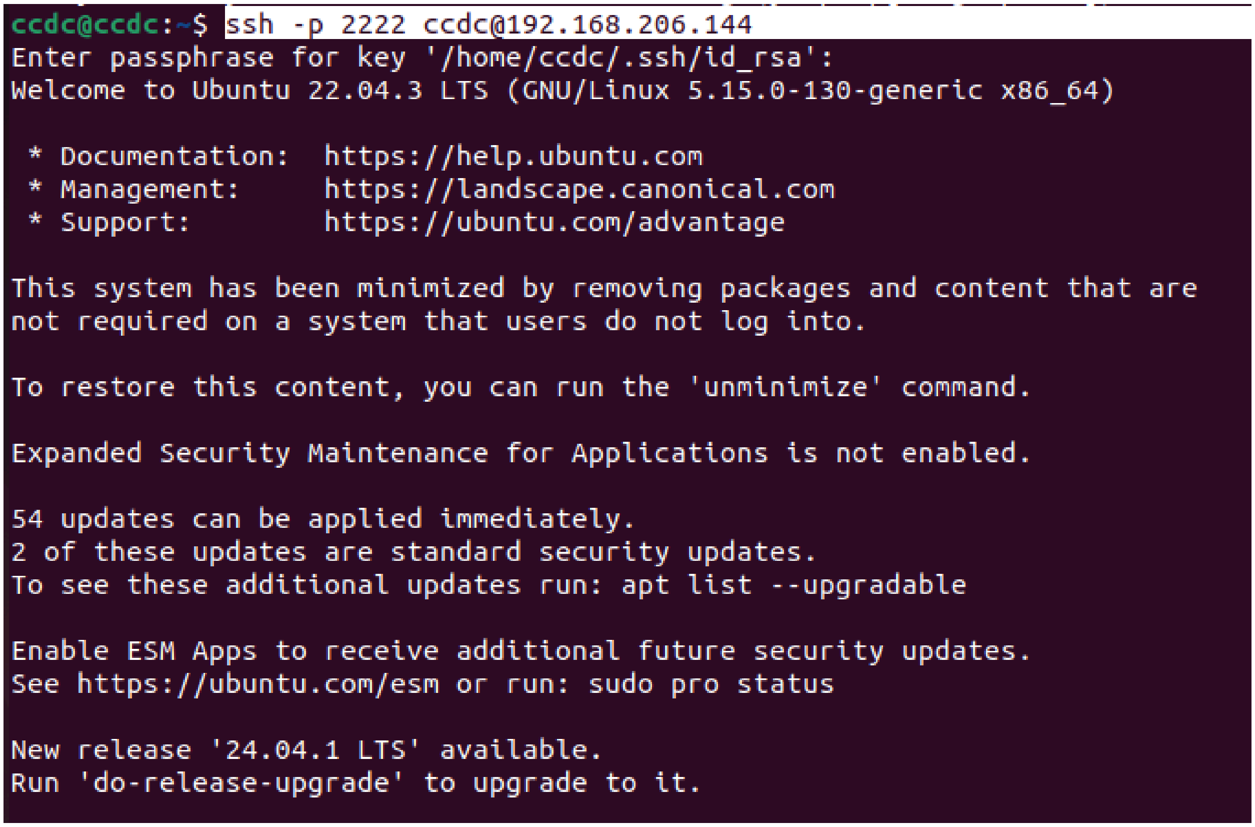
```
ccdc@ccdc:~$ sudo ufw delete allow 22
Rule deleted
Rule deleted (v6)
ccdc@ccdc:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
CUPS                       ALLOW       Anywhere
2222/tcp                   ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
CUPS (v6)                  ALLOW       Anywhere (v6)
2222/tcp (v6)              ALLOW       Anywhere (v6)
```

Then restarted the SSH service

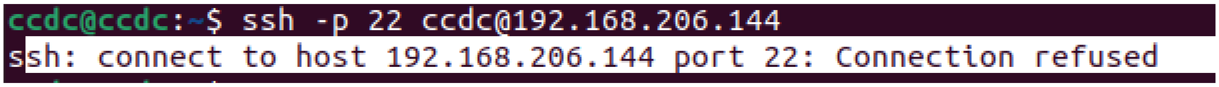


```
ccdc@ccdc:~$ sudo systemctl restart ssh
```

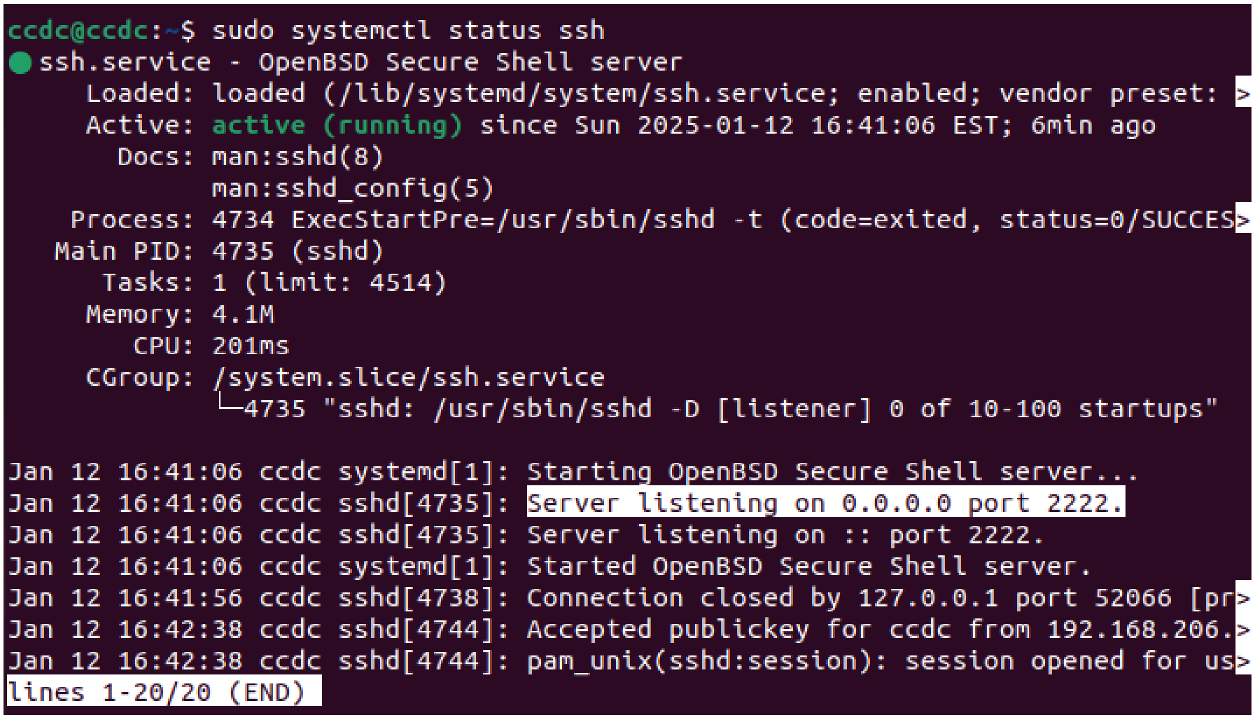
Tested the updated port by logging into port 2222 and 22



```
ccdc@ccdc:~$ ssh -p 2222 ccdc@192.168.206.144
Enter passphrase for key '/home/ccdc/.ssh/id_rsa':
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-130-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

Expanded Security Maintenance for Applications is not enabled.

54 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```



```
ccdc@ccdc:~$ ssh -p 22 ccdc@192.168.206.144
ssh: connect to host 192.168.206.144 port 22: Connection refused
```



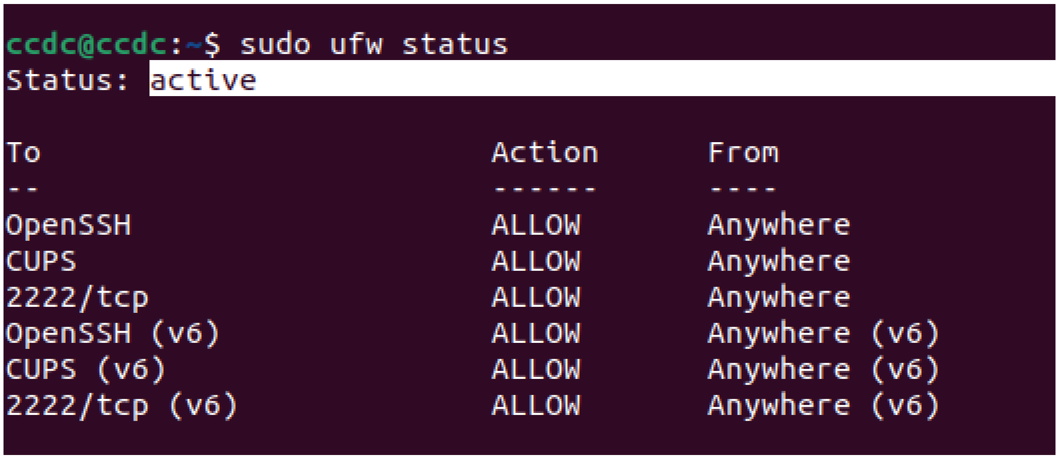
```
ccdc@ccdc:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: >
     Active: active (running) since Sun 2025-01-12 16:41:06 EST; 6min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 4734 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCES>
   Main PID: 4735 (sshd)
      Tasks: 1 (limit: 4514)
     Memory: 4.1M
        CPU: 201ms
     CGroup: /system.slice/ssh.service
             └─4735 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Jan 12 16:41:06 ccdc systemd[1]: Starting OpenBSD Secure Shell server...
Jan 12 16:41:06 ccdc sshd[4735]: Server listening on 0.0.0.0 port 2222.
Jan 12 16:41:06 ccdc sshd[4735]: Server listening on :: port 2222.
Jan 12 16:41:06 ccdc systemd[1]: Started OpenBSD Secure Shell server.
Jan 12 16:41:56 ccdc sshd[4738]: Connection closed by 127.0.0.1 port 52066 [pr>
Jan 12 16:42:38 ccdc sshd[4744]: Accepted publickey for ccdc from 192.168.206.>
Jan 12 16:42:38 ccdc sshd[4744]: pam_unix(sshd:session): session opened for us>
lines 1-20/20 (END)
```

Changing SSH port tutorial

Problem: We want to restrict ssh to a certain IP address so that the only way anyone can access the SSH for our ubuntu machine is to be on a certain IP address.

We can achieve this with UFS

Following this guidance, first we can see UFW is active



```
ccdc@ccdc:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
CUPS                       ALLOW       Anywhere
2222/tcp                   ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
CUPS (v6)                  ALLOW       Anywhere (v6)
2222/tcp (v6)              ALLOW       Anywhere (v6)
```

(to be continued)