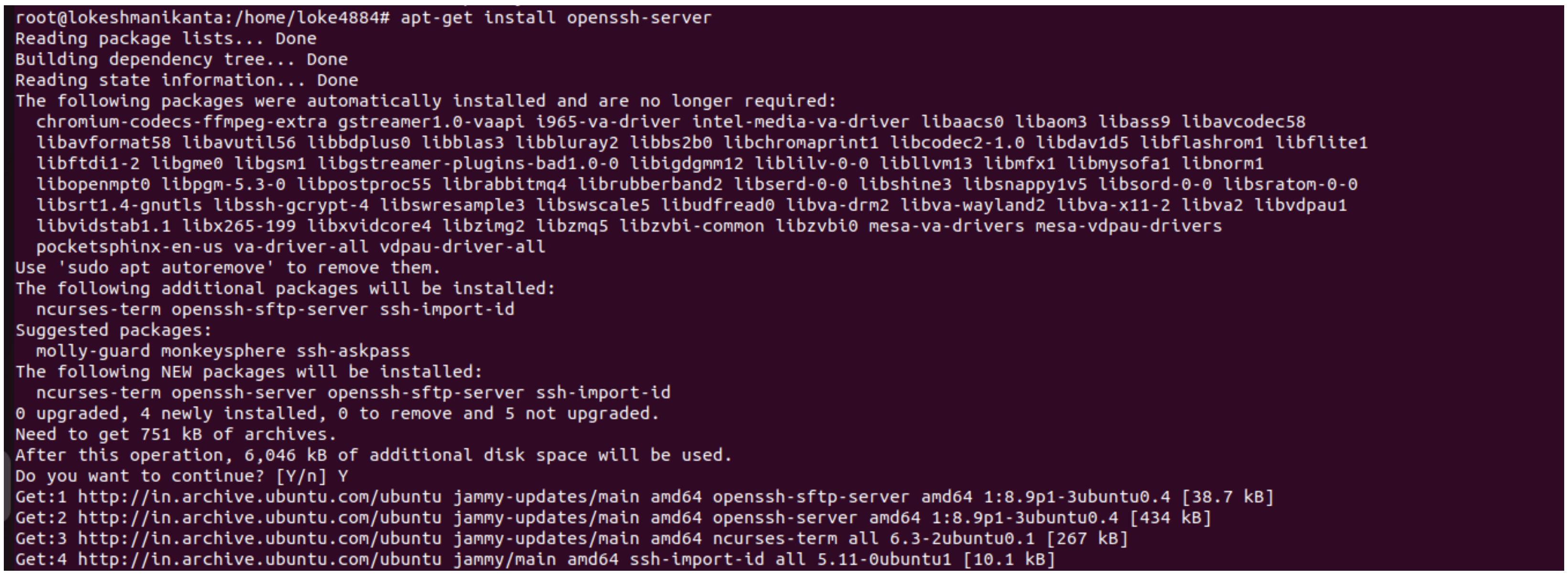
Securing Linux

ssh access from windows

install ssh in ubuntu

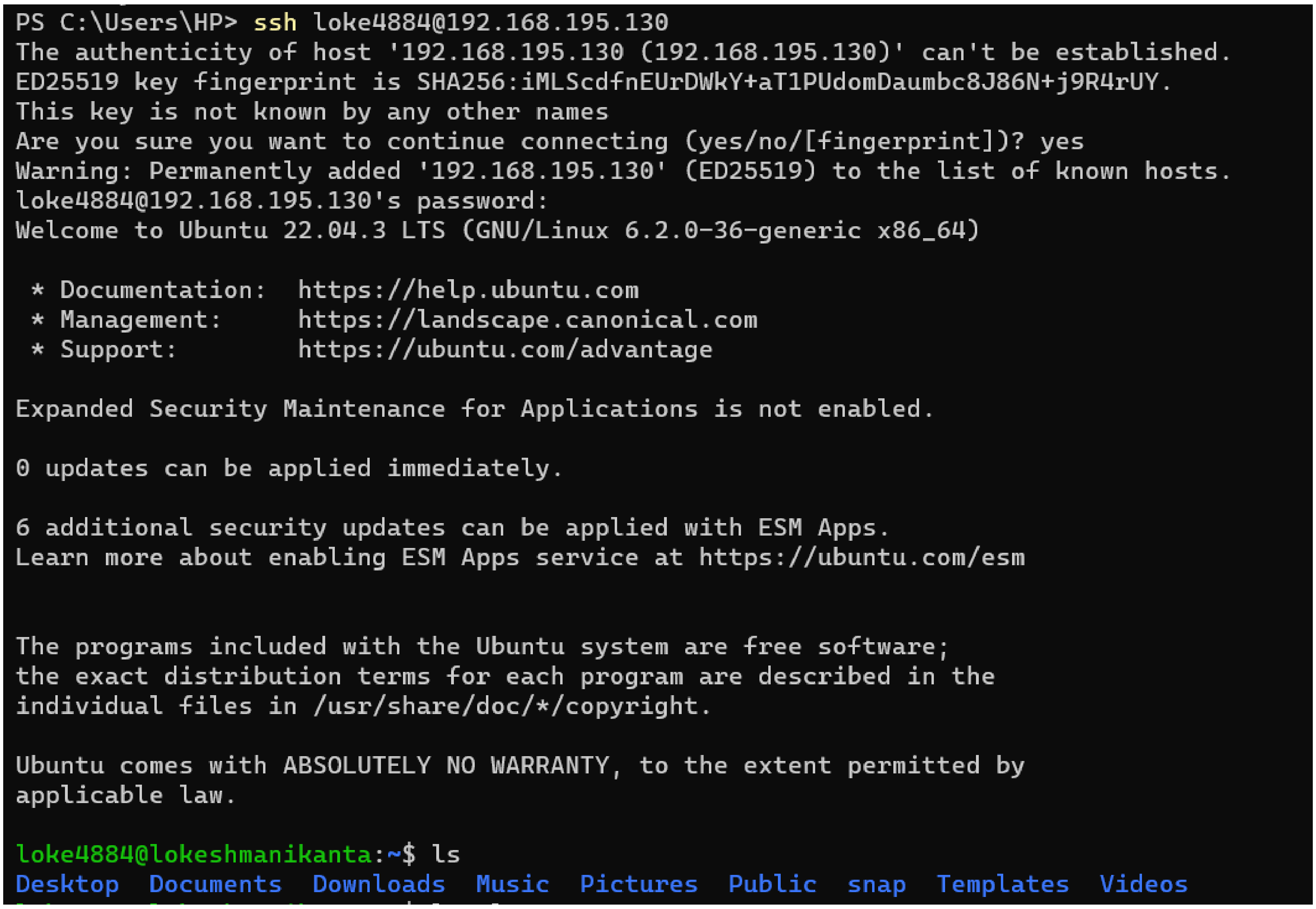
apt-get install openssh-server



```
root@lokeshmanikanta:/home/loke4884# apt-get install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58
  libavformat58 libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1 libcodec2-1.0 libdav1d5 libflashrom1 libflite1
  libftdi1-2 libgme0 libgsm1 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1 libmysofa1 libnorm1
  libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0
  libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1
  libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers mesa-vdpau-drivers
  pocketsphinx-en-us va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 5 not upgraded.
Need to get 751 kB of archives.
After this operation, 6,046 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-sftp-server amd64 1:8.9p1-3ubuntu0.4 [38.7 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 openssh-server amd64 1:8.9p1-3ubuntu0.4 [434 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ncurses-term all 6.3-2ubuntu0.1 [267 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 ssh-import-id all 5.11-0ubuntu1 [10.1 kB]
```

Accessing Linux from windows

ssh username@ip

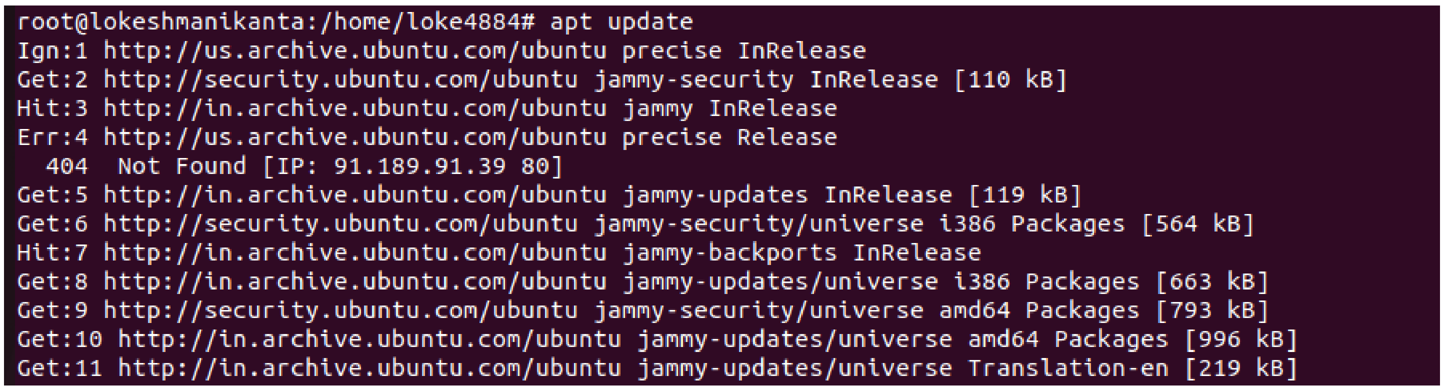


```
PS C:\Users\HP> ssh loke4884@192.168.195.130
The authenticity of host '192.168.195.130 (192.168.195.130)' can't be established.
ED25519 key fingerprint is SHA256:iMLScdfnEUrDWkY+aT1PUdomDaumbc8J86N+j9R4rUY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.195.130' (ED25519) to the list of known hosts.
loke4884@192.168.195.130's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

loke4884@lokeshmanikanta:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Videos
```

1.Atomatic Update

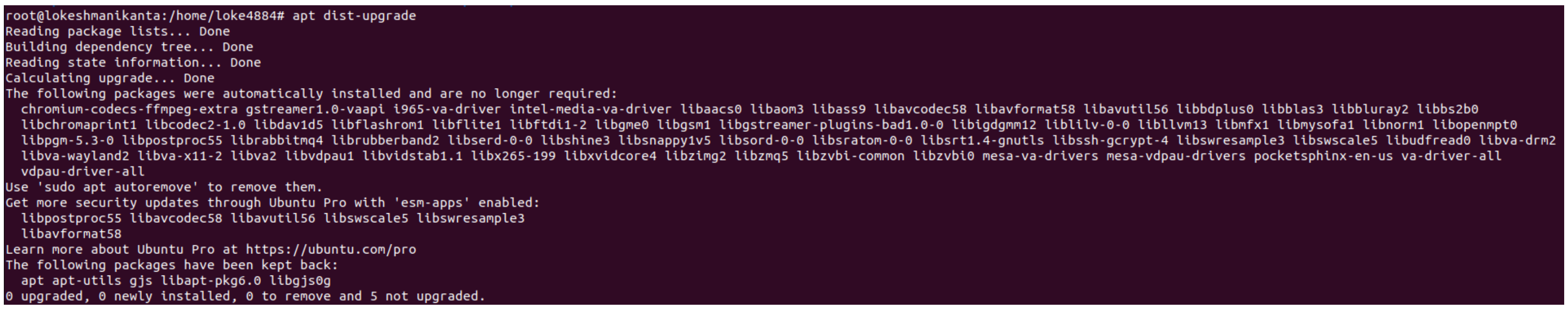
To update manually

sudo apt update



```
root@lokeshmanikanta:/home/loke4884# apt update
Ign:1 http://us.archive.ubuntu.com/ubuntu precise InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Err:4 http://us.archive.ubuntu.com/ubuntu precise Release
  404  Not Found [IP: 91.189.91.39 80]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [564 kB]
Hit:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [663 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [793 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [996 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [219 kB]
```

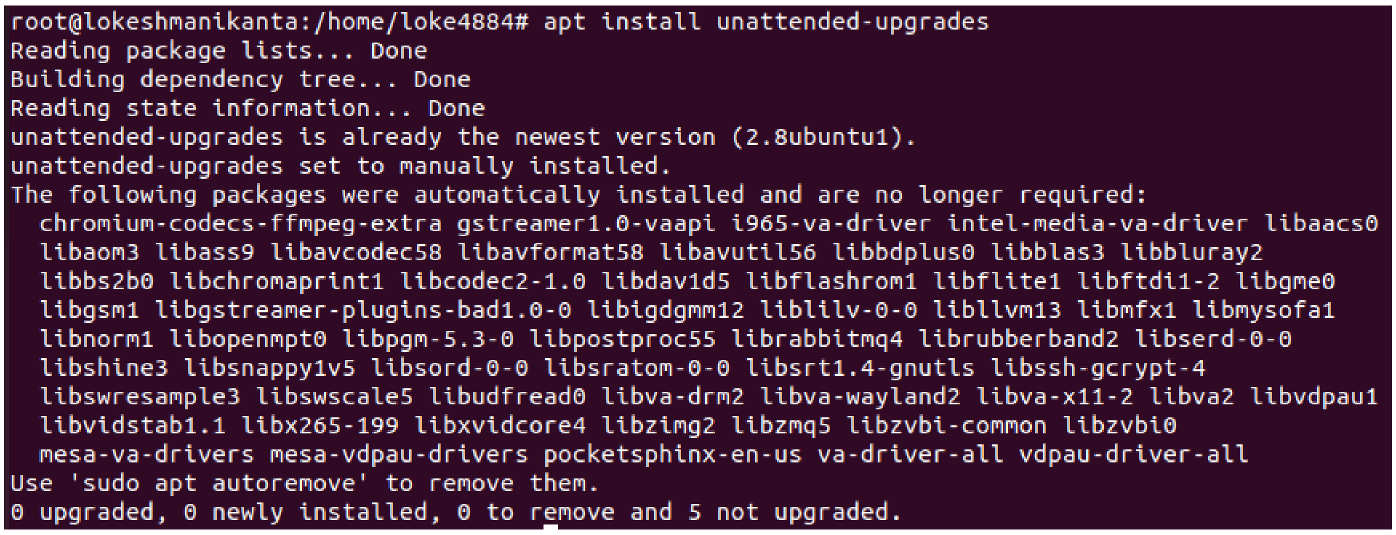
If you want to perform a distribution upgrade (which upgrades the entire operating system to the latest release), you would use:



```
root@lokeshmanikanta:/home/loke4884# apt dist-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0
  libchromaprint1 libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1 libmysofa1 libnorm1 libopenmpt0
  libpgm-5.3-0 libpostproc55 librabbitmq4 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0 libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0 libva-drm2
  libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us va-driver-all
  vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libpostproc55 libavcodec58 libavutil56 libswscale5 libswresample3
  libavformat58
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following packages have been kept back:
  apt apt-utils gjs libapt-pkg6.0 libgjs0g
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

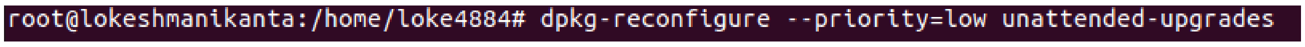
For automatic updates

sudo apt install unattended-upgrades



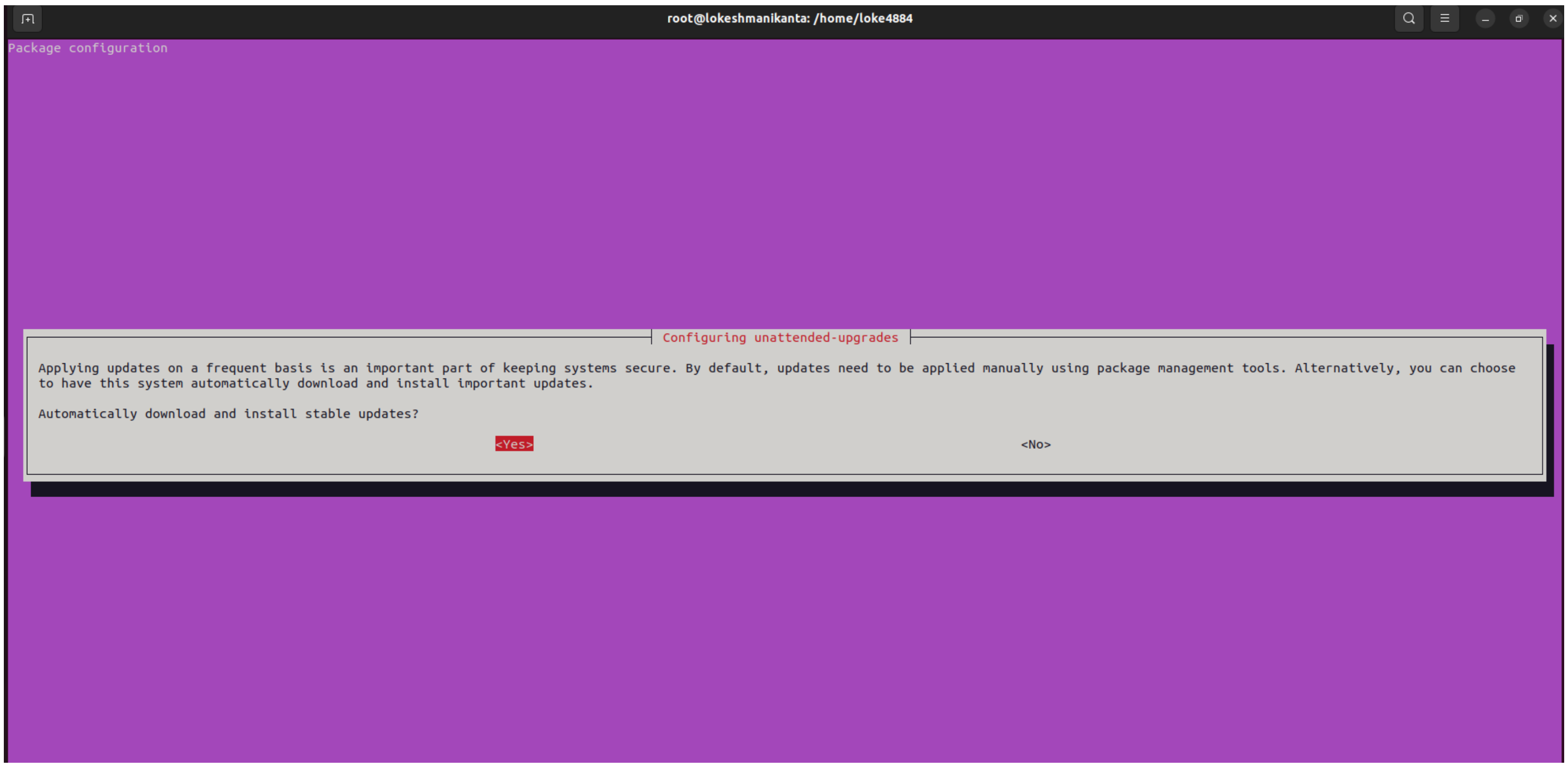
```
root@lokeshmanikanta:/home/loke4884# apt install unattended-upgrades
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unattended-upgrades is already the newest version (2.8ubuntu1).
unattended-upgrades set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0
  libaom3 libass9 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3 libbluray2
  libbs2b0 libchromaprint1 libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0
  libgsm1 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1 libmysofa1
  libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4 librubberband2 libserd-0-0
  libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0 libsrt1.4-gnutls libssh-gcrypt-4
  libswresample3 libswscale5 libudfread0 libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1
  libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0
  mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

dpkg-reconfigure --priority=low unattended-upgrades



```
root@lokeshmanikanta:/home/loke4884# dpkg-reconfigure --priority=low unattended-upgrades
```

Give yes this for automatically updates



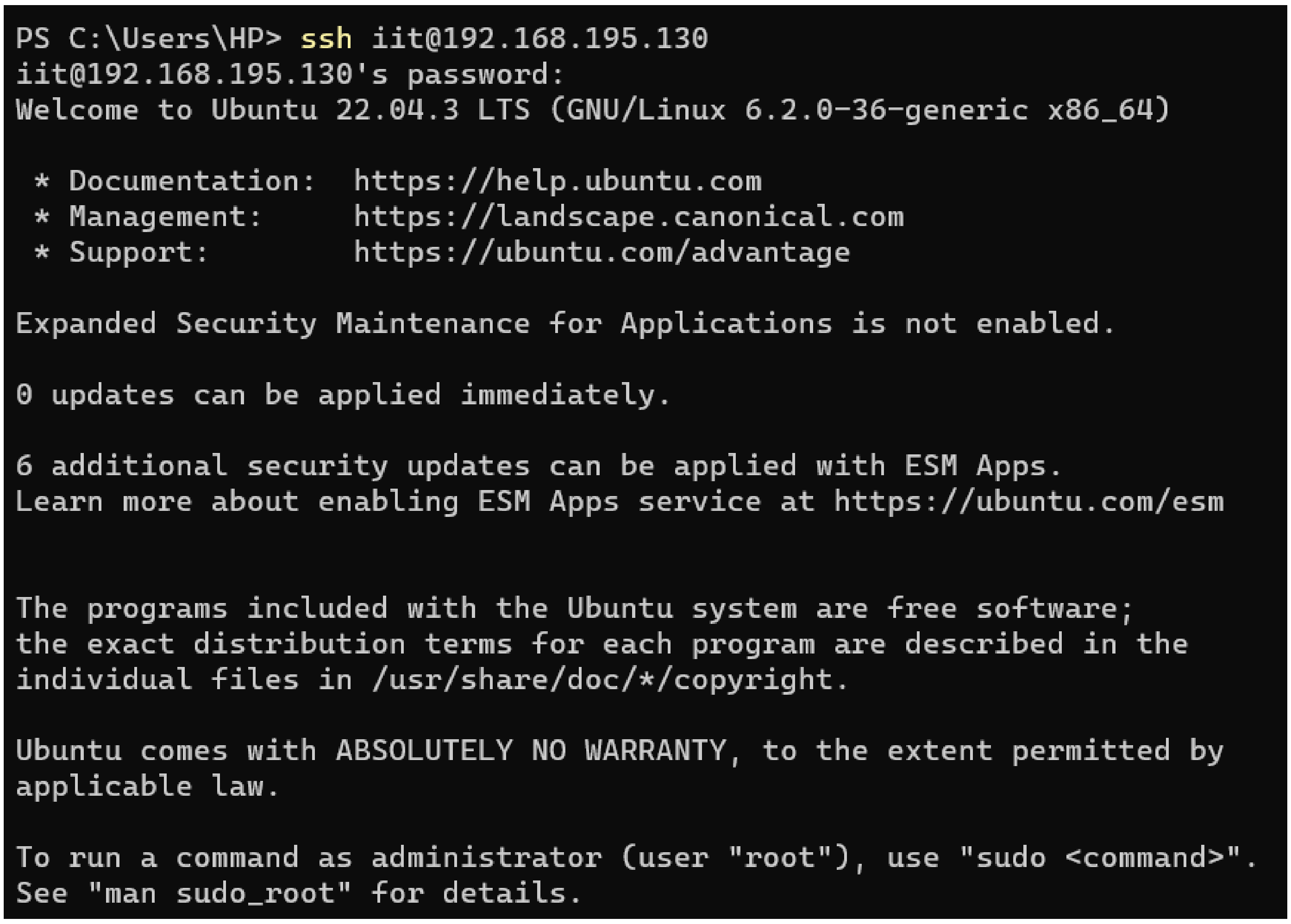
```
Package configuration

                 Configuring unattended-upgrades

Applying updates on a frequent basis is an important part of keeping systems secure. By default, updates need to be applied manually using package management tools. Alternatively, you can choose
to have this system automatically download and install important updates.

Automatically download and install stable updates?

                    <Yes>                                        <No>
```

Passwords for sockers

Creating public and private keys

Public key is assigned for linux server,To unlock that public key we use private key as Authentication Key pairs

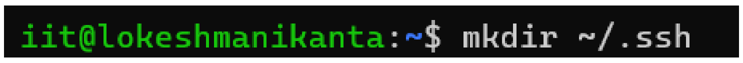
As of now ssh into another user



```
PS C:\Users\HP> ssh iit@192.168.195.130
iit@192.168.195.130's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Private-Public Key pair

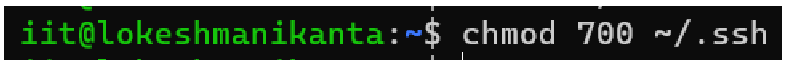
At the root directory made the folder named .ssh which stores all the public keys



```
iit@lokeshmanikanta:~$ mkdir ~/.ssh
```

Giving the permissions to that directory

Here chmod 700 says that owner has full access to the file read,write,execute



```
iit@lokeshmanikanta:~$ chmod 700 ~/.ssh
```

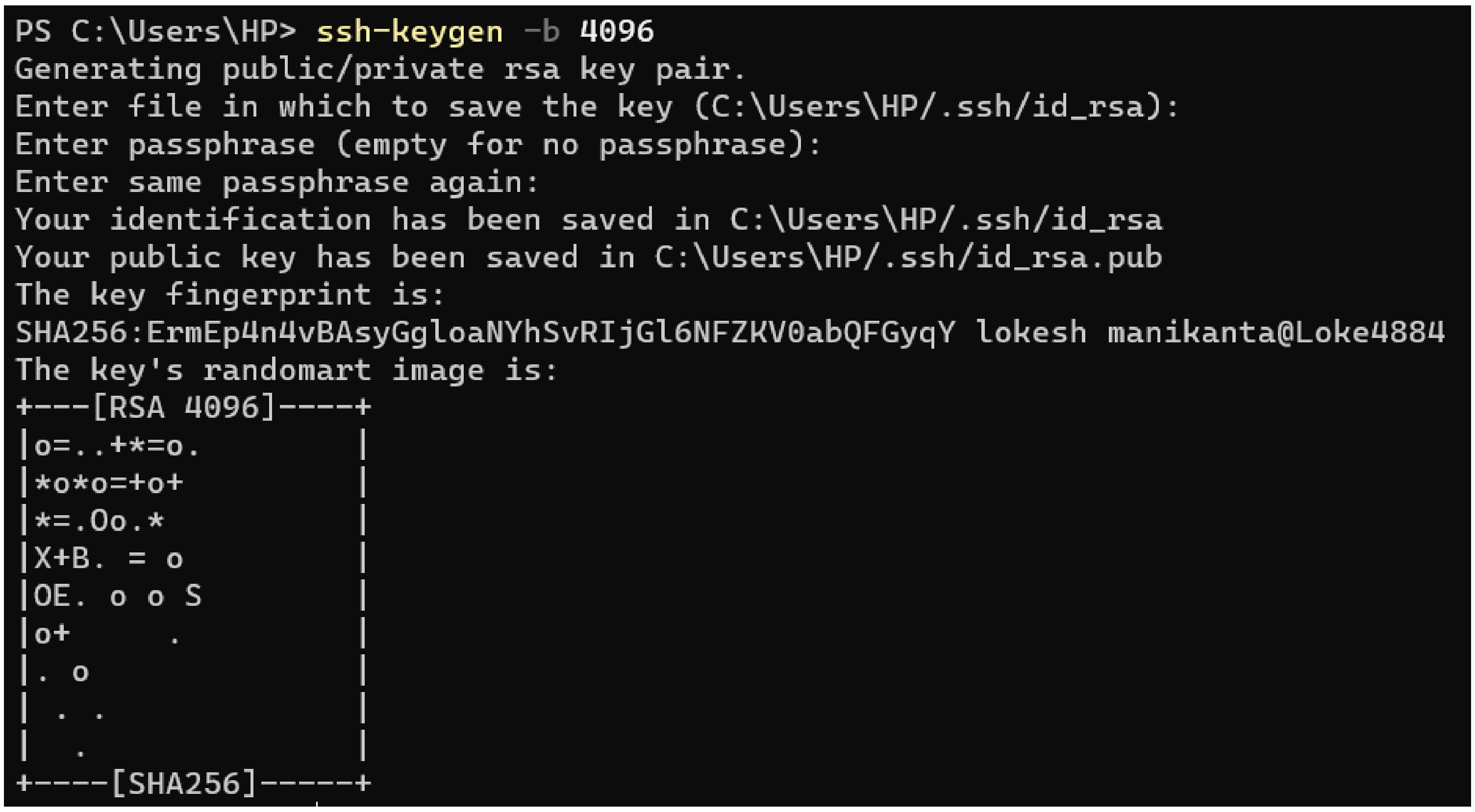
Creating and generating public / private key

ssh-keygen -b 4096

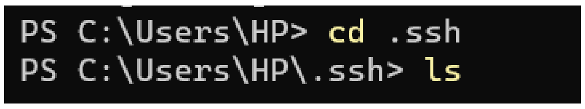
ssh → Create Public/Private Keys

-b → How the big key should be

4096 → Size for the key

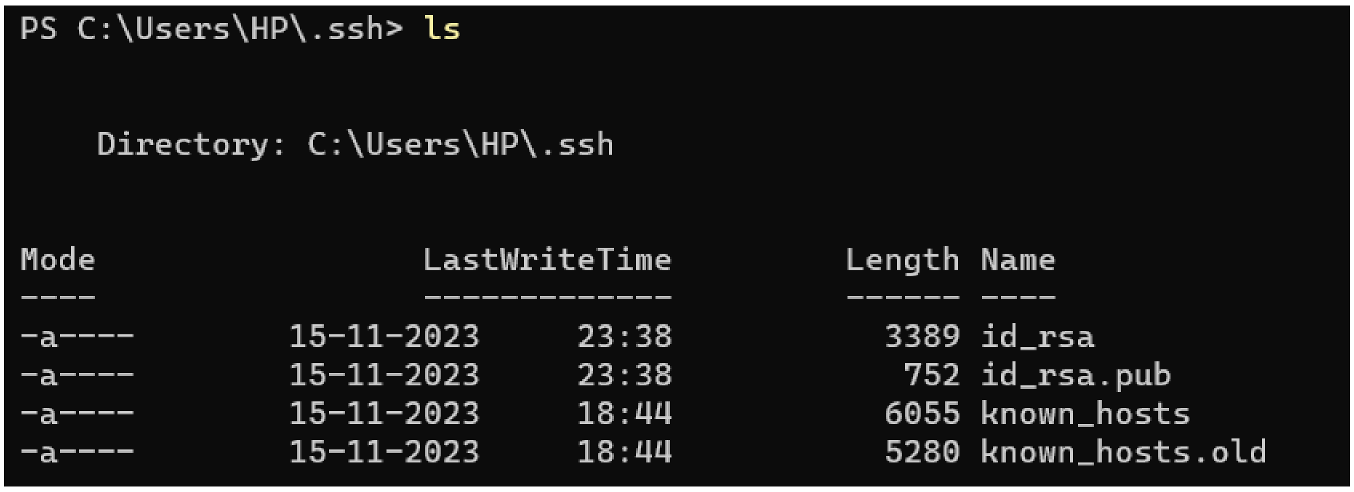


```
PS C:\Users\HP> ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\HP/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\HP/.ssh/id_rsa
Your public key has been saved in C:\Users\HP/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ErmEp4n4vBAsyGgloaNYhSvRIjGl6NFZKV0abQFGyqY lokesh manikanta@Loke4884
The key's randomart image is:
+---[RSA 4096]----+
|o=..+*=o.        |
|*o*o=+o+         |
|*=.Oo.*          |
|X+B. = o         |
|OE. o o S        |
|o+     .         |
|. o              |
| . .             |
|  .              |
+----[SHA256]-----+
```



```
PS C:\Users\HP> cd .ssh
PS C:\Users\HP\.ssh> ls
```

Private- and public keys



```
PS C:\Users\HP\.ssh> ls


    Directory: C:\Users\HP\.ssh


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        15-11-2023     23:38           3389 id_rsa
-a----        15-11-2023     23:38            752 id_rsa.pub
-a----        15-11-2023     18:44           6055 known_hosts
-a----        15-11-2023     18:44           5280 known_hosts.old
```

Assigning public key to the server side

By windows

scp $env:USERPROFILE/.ssh/id_rsa.pub username@serverip:~/.ssh/authorized_keys

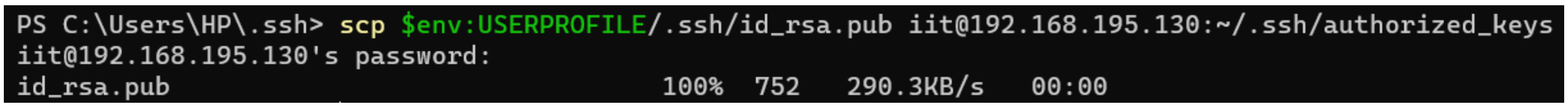
ex: scp $env:USERPROFILE/.ssh/id_rsa.pub iit@192.168.195.130:~/.ssh/authorized_keys

scp → secure copy ,Command

command to copy the public key file (id_rsa.pub) from the .ssh directory in your local user profile on a Windows system to the .ssh directory on a remote server.

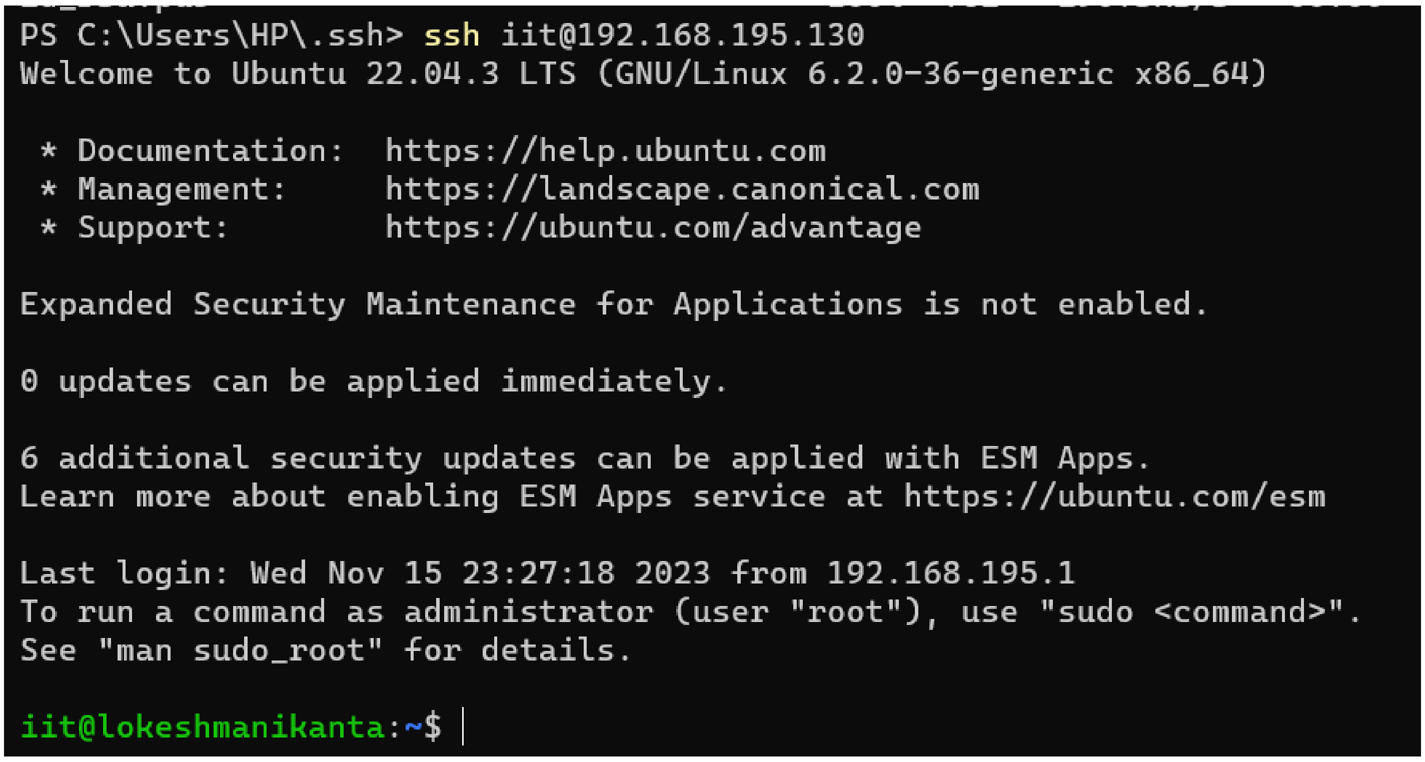
$env:USERPROFILE → This is an environment variable in PowerShell on Windows. It represents the path to the current user's profile directory. It is similar to using ~/ on Unix-like systems to refer to the home directory.

scp $env:USERPROFILE/.ssh/id_rsa.pub username@serverip:~/.ssh/authorized_keys -->command is copying your local public key file to the authorized_keys file on a remote server, allowing you to authenticate to that server using your private key. This is a common step in setting up SSH key-based authentication for secure and convenient access to remote servers.



```
PS C:\Users\HP\.ssh> scp $env:USERPROFILE/.ssh/id_rsa.pub iit@192.168.195.130:~/.ssh/authorized_keys
iit@192.168.195.130's password:
id_rsa.pub                                      100%  752    290.3KB/s   00:00
```

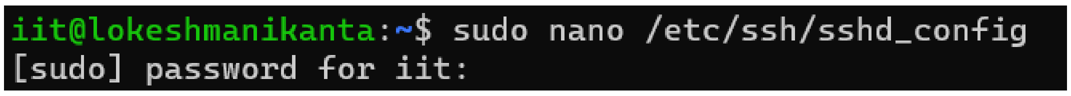
Now without any password Windows can able to access the server(ubuntu machine) as of windows has private key to public key of ubuntu so based on public-private key pair for linux access from another machine can be done to more secure



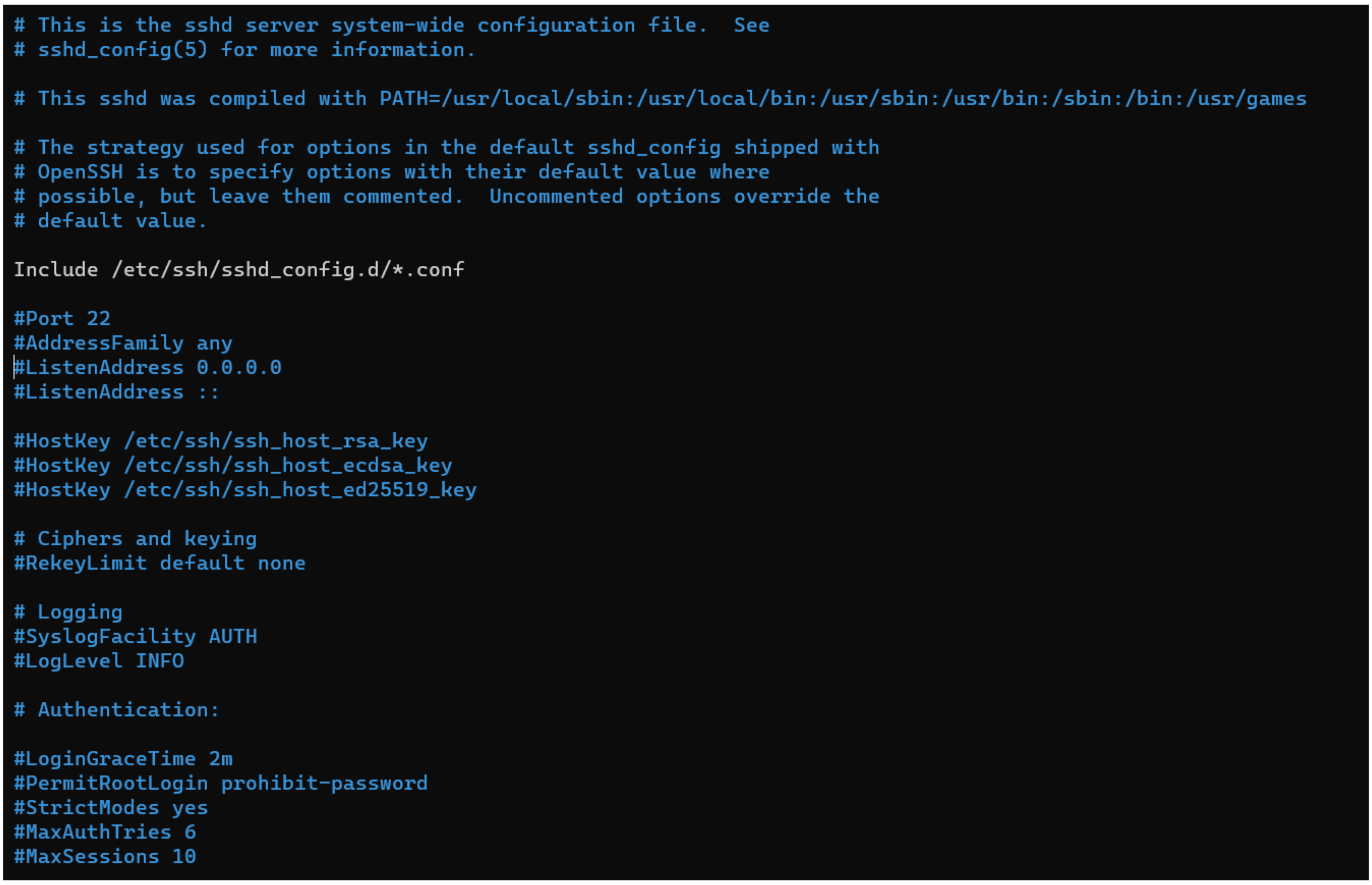
```
PS C:\Users\HP\.ssh> ssh iit@192.168.195.130
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Wed Nov 15 23:27:18 2023 from 192.168.195.1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

iit@lokeshmanikanta:~$
```

Going into ssh configuration file

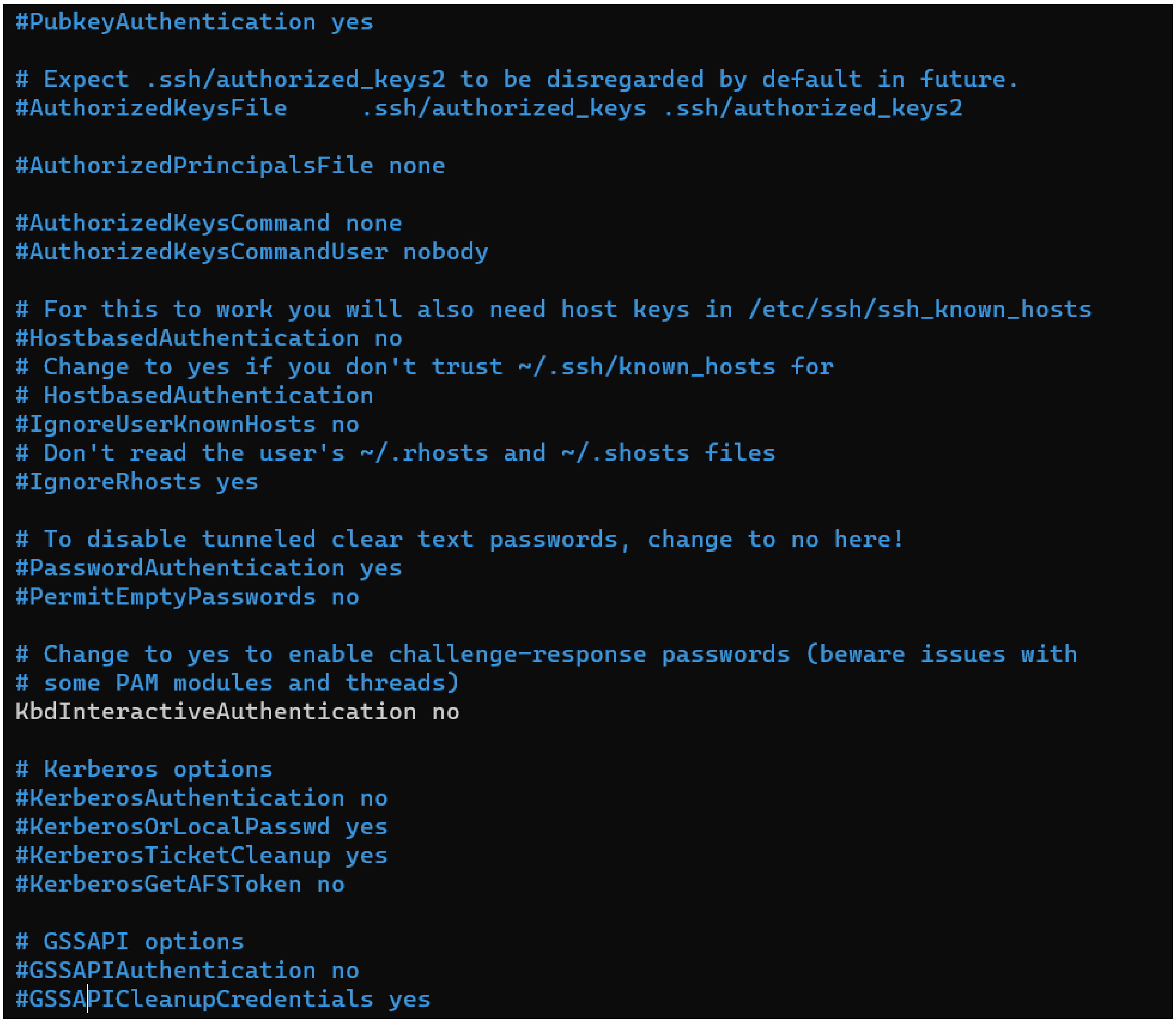
sudo nano /etc/ssh/sshd_config



```
iit@lokeshmanikanta:~$ sudo nano /etc/ssh/sshd_config
[sudo] password for iit:
```



```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

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

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```



```
#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
```

To increase much security

You can assign port number so that default port number is changed to custom port number to access ssh

For ex :

Port 22 is changed to Port 712

To access the ssh

ssh username@server-ip -p portnumber

Address Family any (to access IPV4 and IPV6)

Address Fmaily inet → You can set it to only IPV4 only

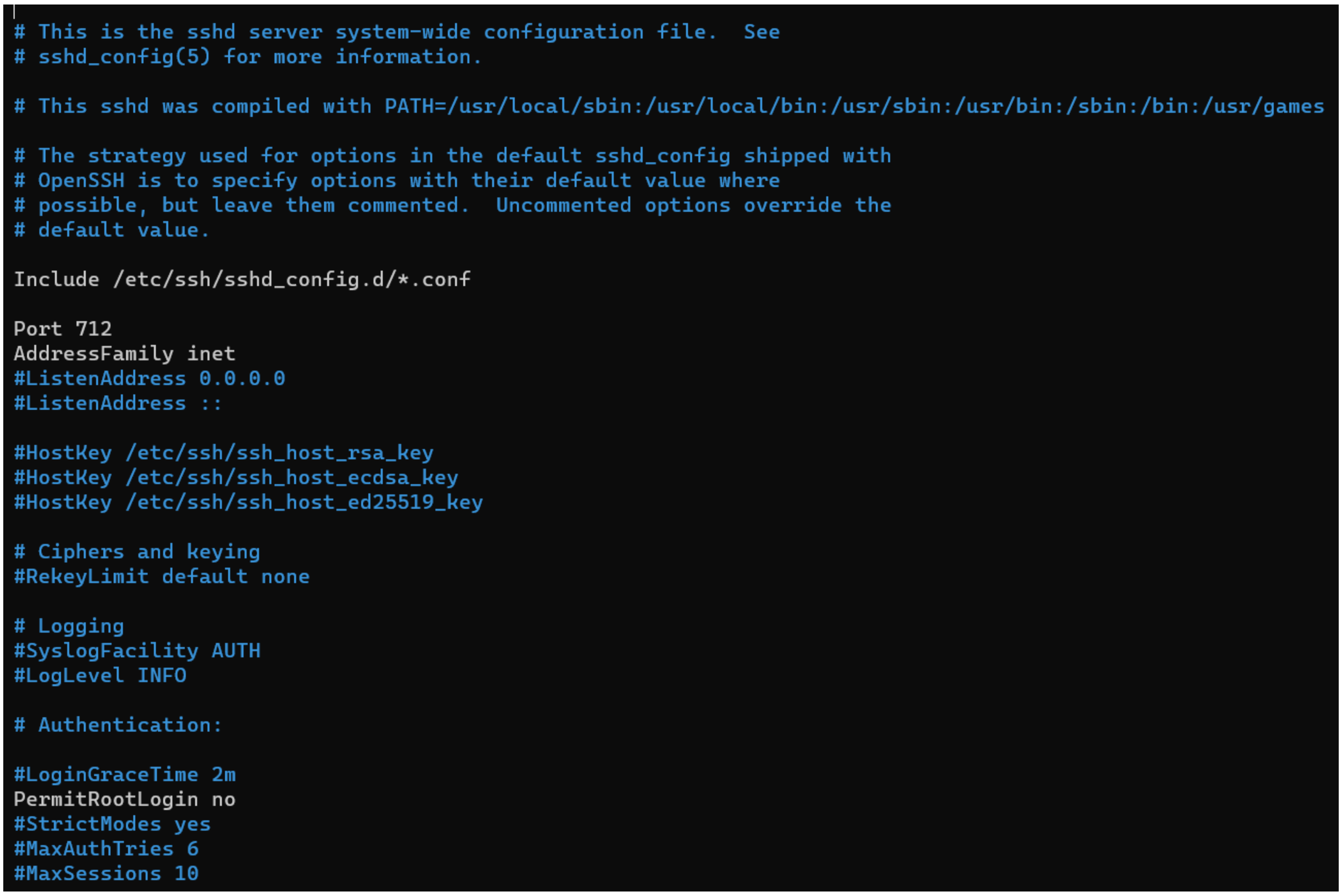
PermitRootLogin no → such that ssh cant access root

For sockers

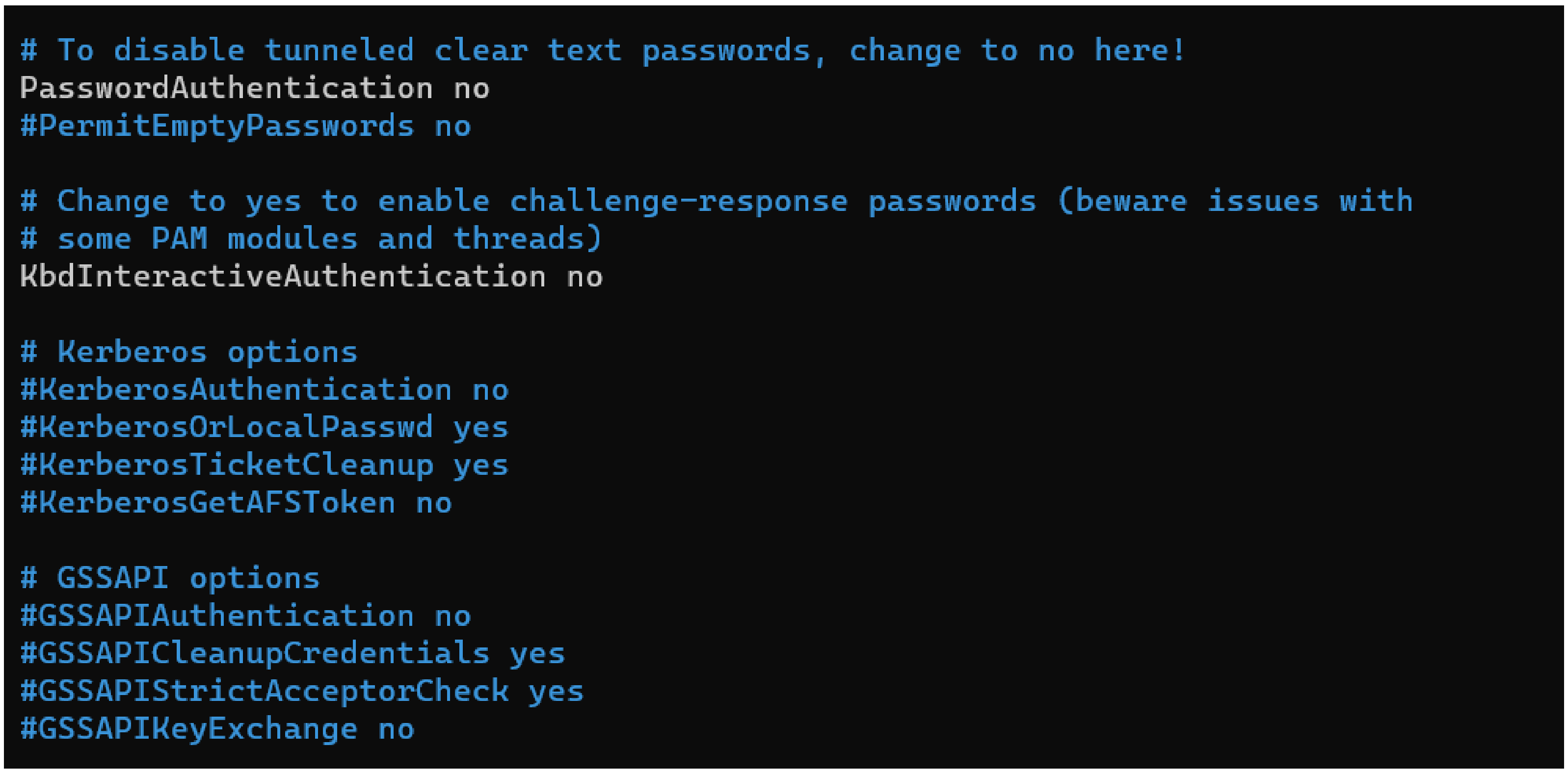
PasswordAuthentication yes → is set to PasswordAuthentication no (so that only public-private key pair authentication takes place which is more secure than PasswordAuthentication) otherise if we left PasswordAuthentication yes then without public-private key pair authentication hacker can access easily

After doing this all changes

sudo systemctl restart sshd

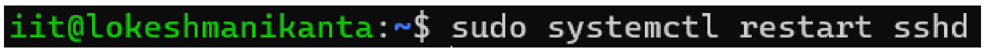


```
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

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

Port 712
AddressFamily inet
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```



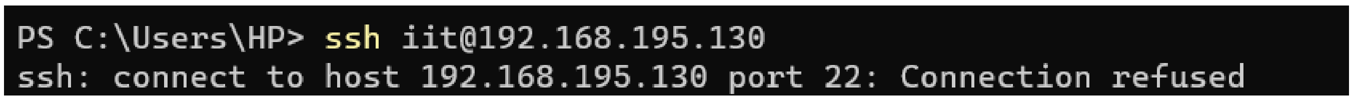
```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no
```

Restart ssh



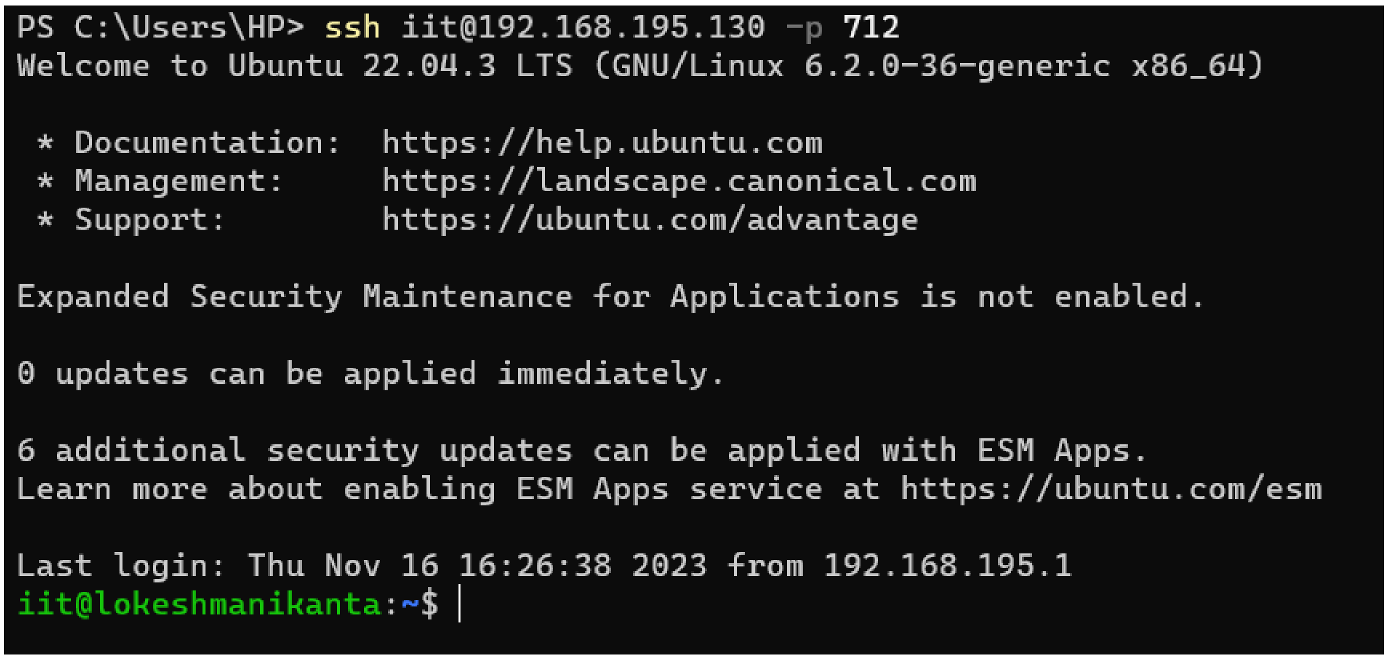
```
iit@lokeshmanikanta:~$ sudo systemctl restart sshd
```

Without the specific port which ssh is assigned you can access linux machine



```
PS C:\Users\HP> ssh iit@192.168.195.130
ssh: connect to host 192.168.195.130 port 22: Connection refused
```

Accessing linux by customised specific port

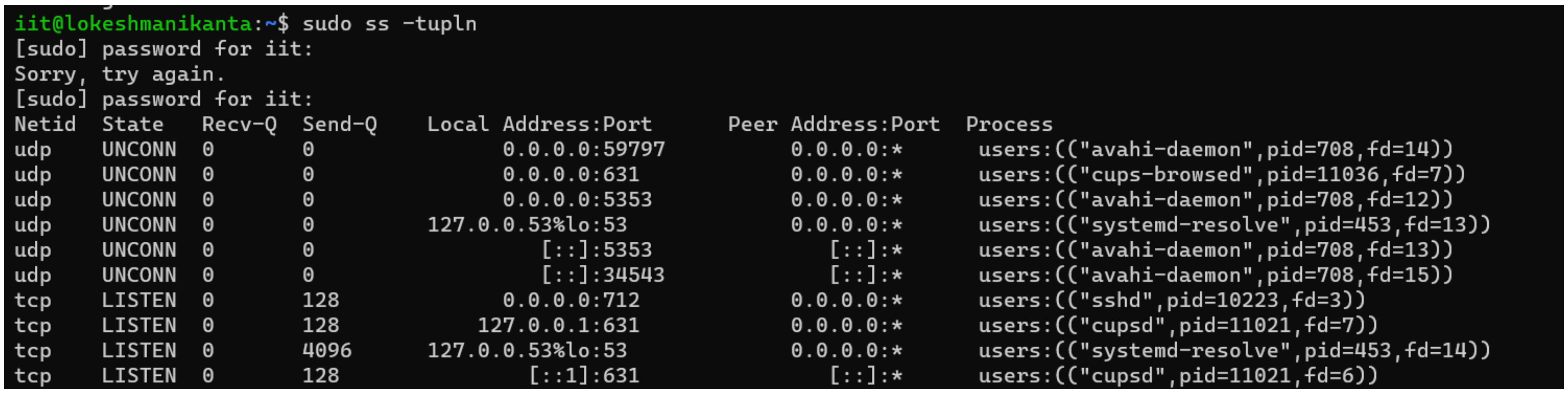


```
PS C:\Users\HP> ssh iit@192.168.195.130 -p 712
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Nov 16 16:26:38 2023 from 192.168.195.1
iit@lokeshmanikanta:~$ 
```

Making more secure such that allowing ssh only through firewall

To see which ports are listening

sudo ss -tupln



```
iit@lokeshmanikanta:~$ sudo ss -tupln
[sudo] password for iit:
Sorry, try again.
[sudo] password for iit:
Netid  State   Recv-Q  Send-Q    Local Address:Port      Peer Address:Port  Process
udp    UNCONN  0       0               0.0.0.0:59797          0.0.0.0:*      users:(("avahi-daemon",pid=708,fd=14))
udp    UNCONN  0       0               0.0.0.0:631            0.0.0.0:*      users:(("cups-browsed",pid=11036,fd=7))
udp    UNCONN  0       0               0.0.0.0:5353           0.0.0.0:*      users:(("avahi-daemon",pid=708,fd=12))
udp    UNCONN  0       0         127.0.0.53%lo:53             0.0.0.0:*      users:(("systemd-resolve",pid=453,fd=13))
udp    UNCONN  0       0                  [::]:5353              [::]:*      users:(("avahi-daemon",pid=708,fd=13))
udp    UNCONN  0       0                  [::]:34543             [::]:*      users:(("avahi-daemon",pid=708,fd=15))
tcp    LISTEN  0       128             0.0.0.0:712            0.0.0.0:*      users:(("sshd",pid=10223,fd=3))
tcp    LISTEN  0       128           127.0.0.1:631            0.0.0.0:*      users:(("cupsd",pid=11021,fd=7))
tcp    LISTEN  0       4096      127.0.0.53%lo:53             0.0.0.0:*      users:(("systemd-resolve",pid=453,fd=14))
tcp    LISTEN  0       128               [::1]:631               [::]:*      users:(("cupsd",pid=11021,fd=6))
```

we can check port ports at local address:Port

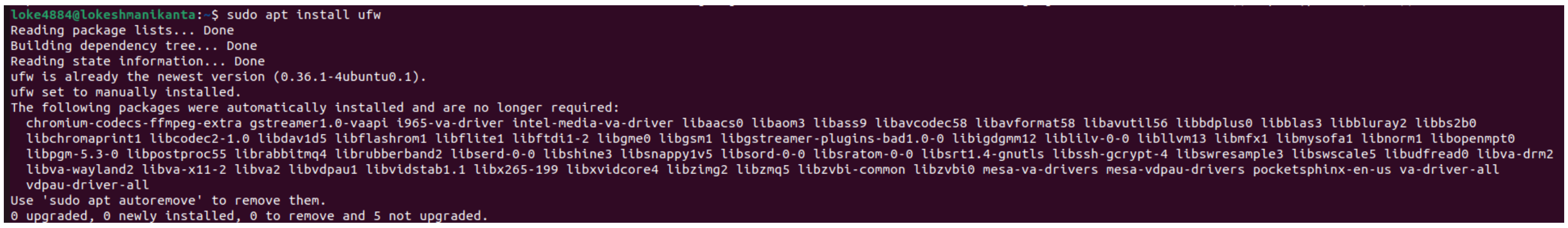
enabling firewall for only ssh access

Setting up the ufw(uncomplicated Firewall):

If not available then install ufw:

Installation of ufw

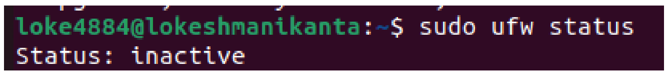
sudo apt install ufw



```
loke4884@lokeshmanikanta:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.1-4ubuntu0.1).
ufw set to manually installed.
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58 libavutil56 libbdplus0 libblas3 libbluray2 libbs2b0
  libchromaprint1 libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1 libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1 libmysofa1 libnorm1 libopenmpt0
  libpgm-5.3-0 libpostproc55 librabbitmq4 librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0 libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0 libva-drm2
  libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers mesa-vdpau-drivers pocketsphinx-en-us va-driver-all
  vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
```

To get activity of ufw

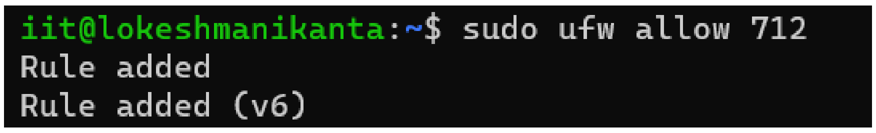
sudo ufw status



```
loke4884@lokeshmanikanta:~$ sudo ufw status
Status: inactive
```

Allowing only ssh port

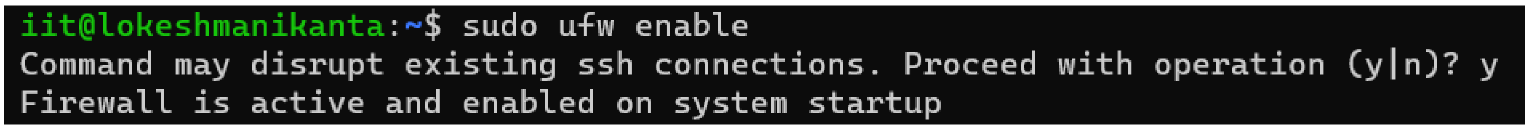
sudo ufw allow 712



```
iit@lokeshmanikanta:~$ sudo ufw allow 712
Rule added
Rule added (v6)
```

Enabling firewall

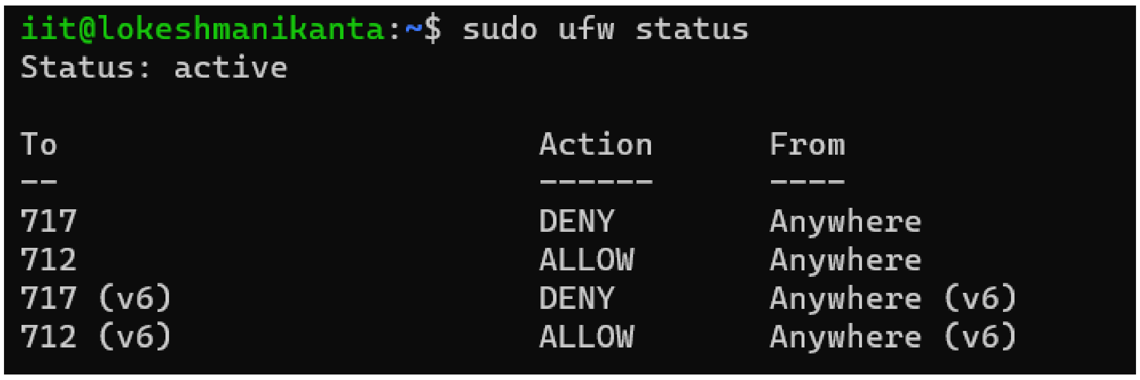
sudo ufw enable



```
iit@lokeshmanikanta:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

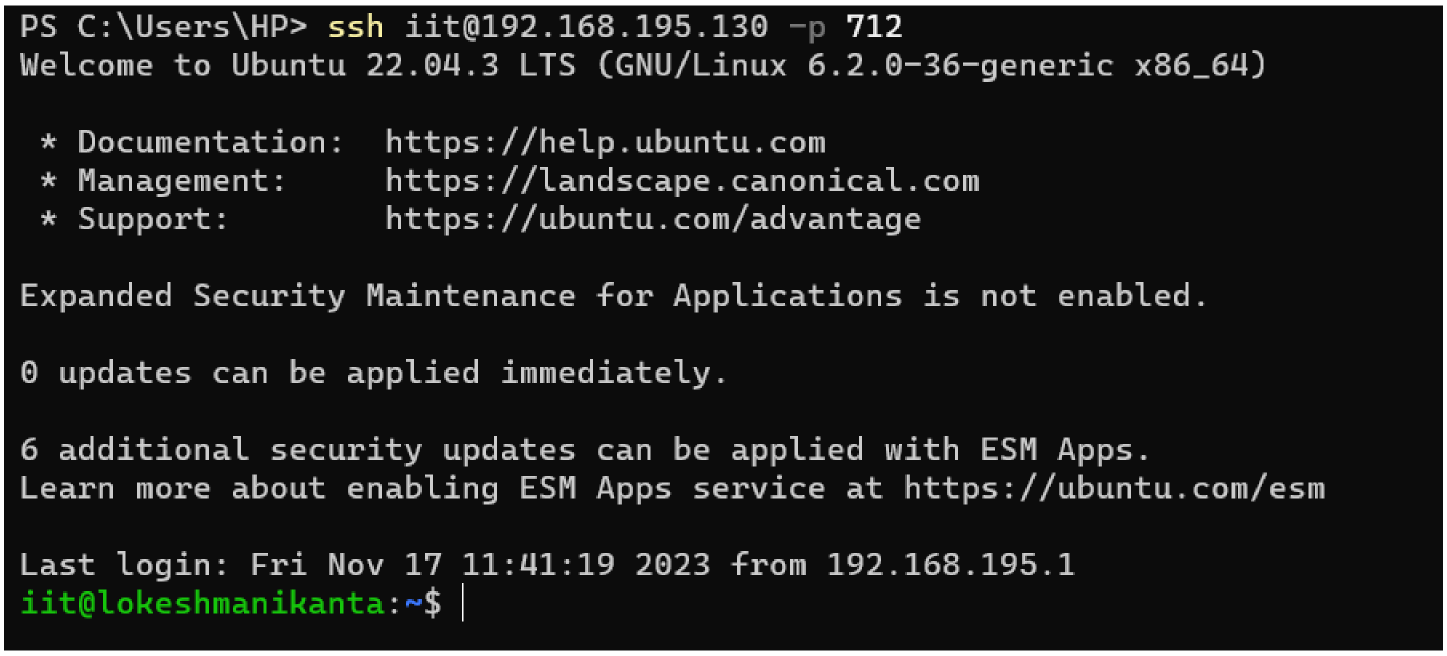
Check the status of the firewall rule

sudo ufw status



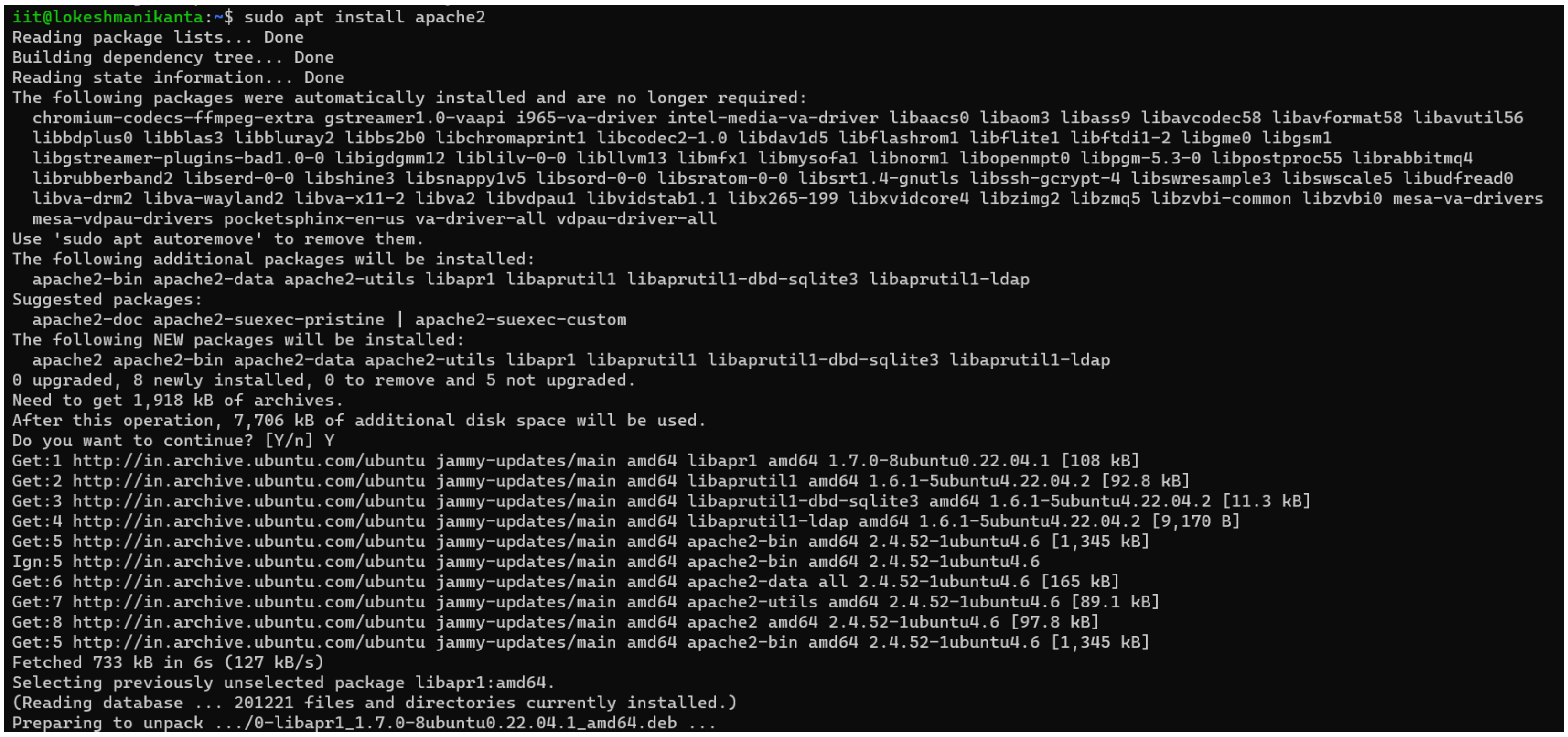
```
iit@lokeshmanikanta:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
717                        DENY        Anywhere
712                        ALLOW       Anywhere
717 (v6)                   DENY        Anywhere (v6)
712 (v6)                   ALLOW       Anywhere (v6)
```

Only ssh through port 712 is enabled*



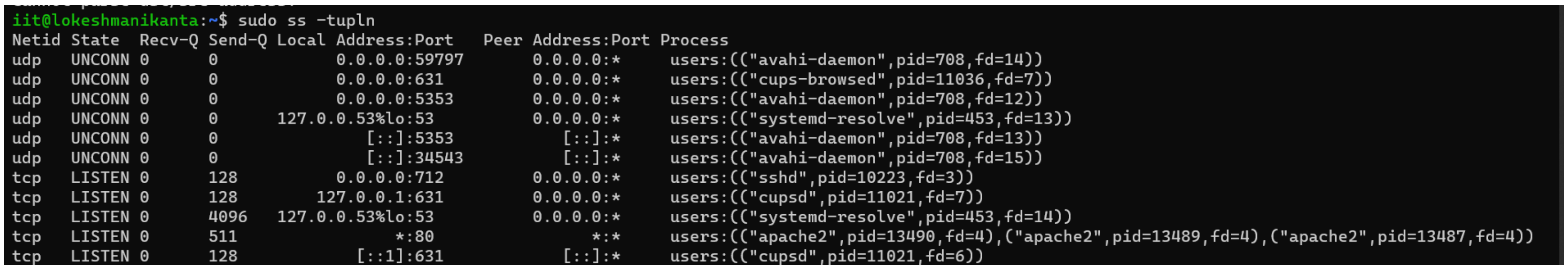
```
PS C:\Users\HP> ssh iit@192.168.195.130 -p 712
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Fri Nov 17 11:41:19 2023 from 192.168.195.1
iit@lokeshmanikanta:~$
```

Suppose as of test you try to use apache2 which is hosted in linux



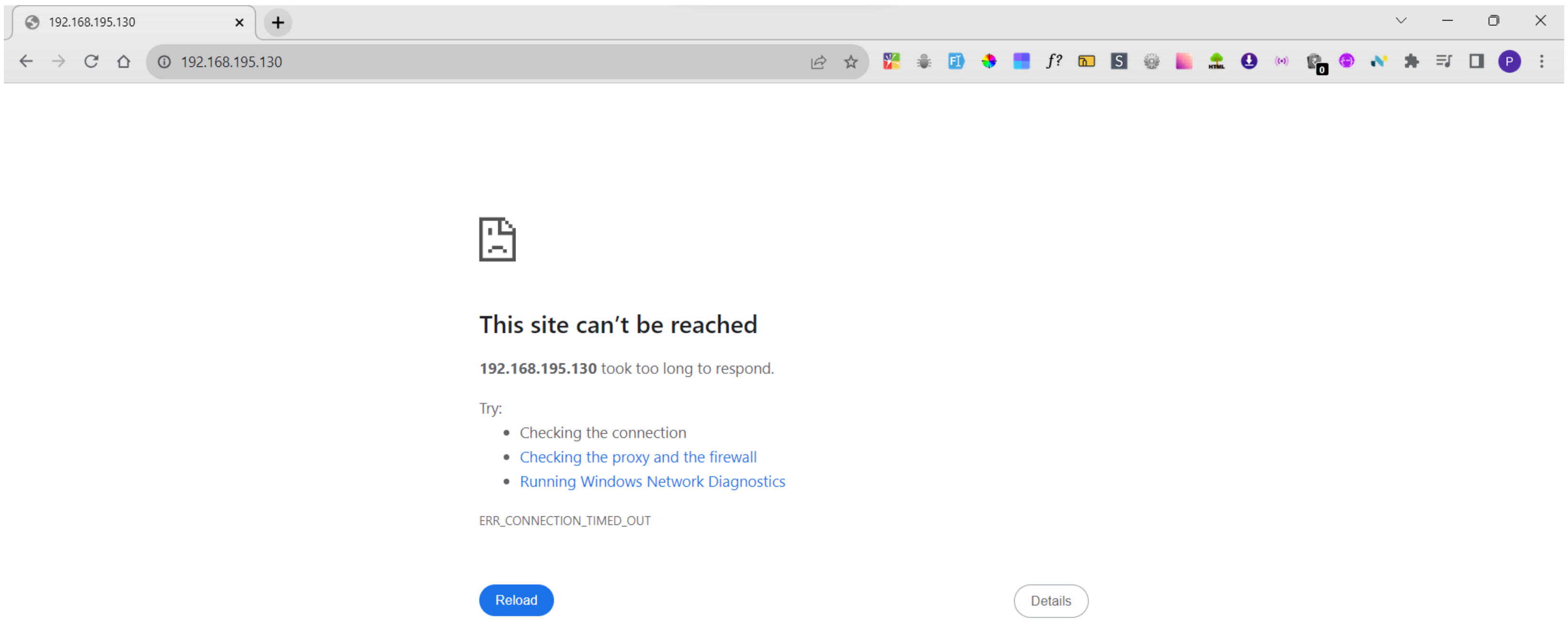
```
iit@lokeshmanikanta:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi i965-va-driver intel-media-va-driver libaacs0 libaom3 libass9 libavcodec58 libavformat58 libavutil56
  libbdplus0 libblas3 libbluray2 libbs2b0 libchromaprint1 libcodec2-1.0 libdav1d5 libflashrom1 libflite1 libftdi1-2 libgme0 libgsm1
  libgstreamer-plugins-bad1.0-0 libigdgmm12 liblilv-0-0 libllvm13 libmfx1 libmysofa1 libnorm1 libopenmpt0 libpgm-5.3-0 libpostproc55 librabbitmq4
  librubberband2 libserd-0-0 libshine3 libsnappy1v5 libsord-0-0 libsratom-0-0 libsrt1.4-gnutls libssh-gcrypt-4 libswresample3 libswscale5 libudfread0
  libva-drm2 libva-wayland2 libva-x11-2 libva2 libvdpau1 libvidstab1.1 libx265-199 libxvidcore4 libzimg2 libzmq5 libzvbi-common libzvbi0 mesa-va-drivers
  mesa-vdpau-drivers pocketsphinx-en-us va-driver-all vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 5 not upgraded.
Need to get 1,918 kB of archives.
After this operation, 7,706 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2 [92.8 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubuntu4.22.04.2 [11.3 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-5ubuntu4.22.04.2 [9,170 B]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.6 [1,345 kB]
Ign:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.6
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.6 [165 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.6 [89.1 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.6 [97.8 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.6 [1,345 kB]
Fetched 733 kB in 6s (127 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 201221 files and directories currently installed.)
Preparing to unpack .../0-libapr1_1.7.0-8ubuntu0.22.04.1_amd64.deb ...
```

Now you can see 80 port is listened



```
iit@lokeshmanikanta:~$ sudo ss -tupln
Netid State  Recv-Q Send-Q Local Address:Port   Peer Address:Port Process
udp   UNCONN 0      0            0.0.0.0:59797       0.0.0.0:*     users:(("avahi-daemon",pid=708,fd=14))
udp   UNCONN 0      0            0.0.0.0:631         0.0.0.0:*     users:(("cups-browsed",pid=11036,fd=7))
udp   UNCONN 0      0            0.0.0.0:5353        0.0.0.0:*     users:(("avahi-daemon",pid=708,fd=12))
udp   UNCONN 0      0      127.0.0.53%lo:53          0.0.0.0:*     users:(("systemd-resolve",pid=453,fd=13))
udp   UNCONN 0      0               [::]:5353           [::]:*     users:(("avahi-daemon",pid=708,fd=13))
udp   UNCONN 0      0               [::]:34543          [::]:*     users:(("avahi-daemon",pid=708,fd=15))
tcp   LISTEN 0      128          0.0.0.0:712         0.0.0.0:*     users:(("sshd",pid=10223,fd=3))
tcp   LISTEN 0      128        127.0.0.1:631         0.0.0.0:*     users:(("cupsd",pid=11021,fd=7))
tcp   LISTEN 0      4096   127.0.0.53%lo:53          0.0.0.0:*     users:(("systemd-resolve",pid=453,fd=14))
tcp   LISTEN 0      511                *:80                *:*     users:(("apache2",pid=13490,fd=4),("apache2",pid=13489,fd=4),("apache2",pid=13487,fd=4))
tcp   LISTEN 0      128            [::1]:631            [::]:*     users:(("cupsd",pid=11021,fd=6))
```

When you try to access apache2 it does not load in another machine(which is in same network) as of firewall is blocking



This site can't be reached

192.168.195.130 took too long to respond.

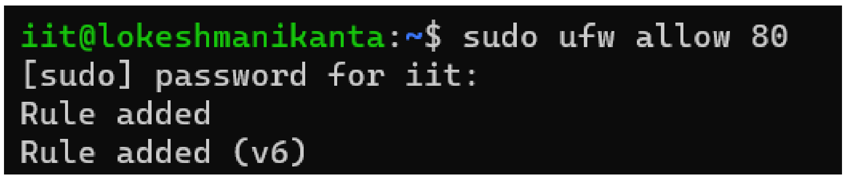
Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

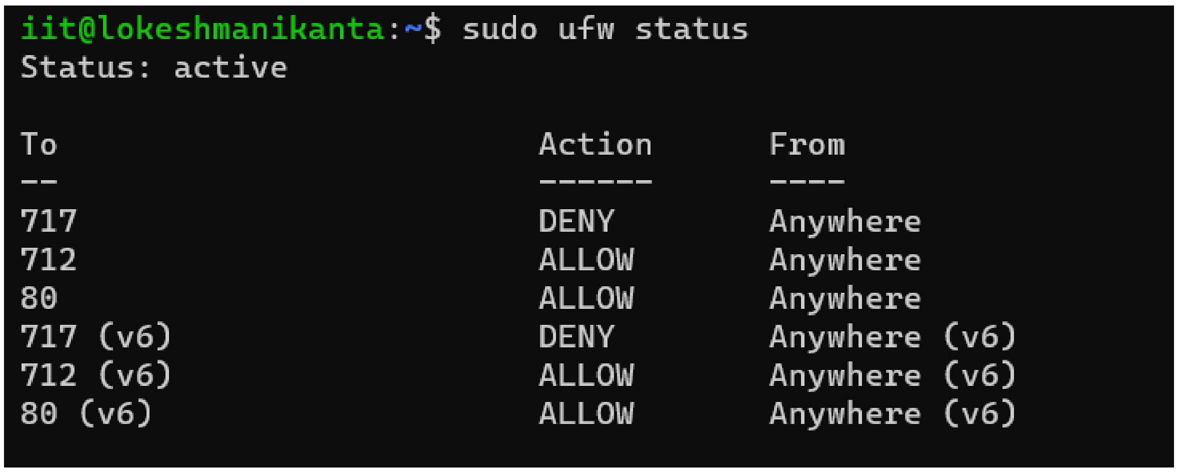
You can also manually allow port 80 in firewall to access

sudo ufw allow 80



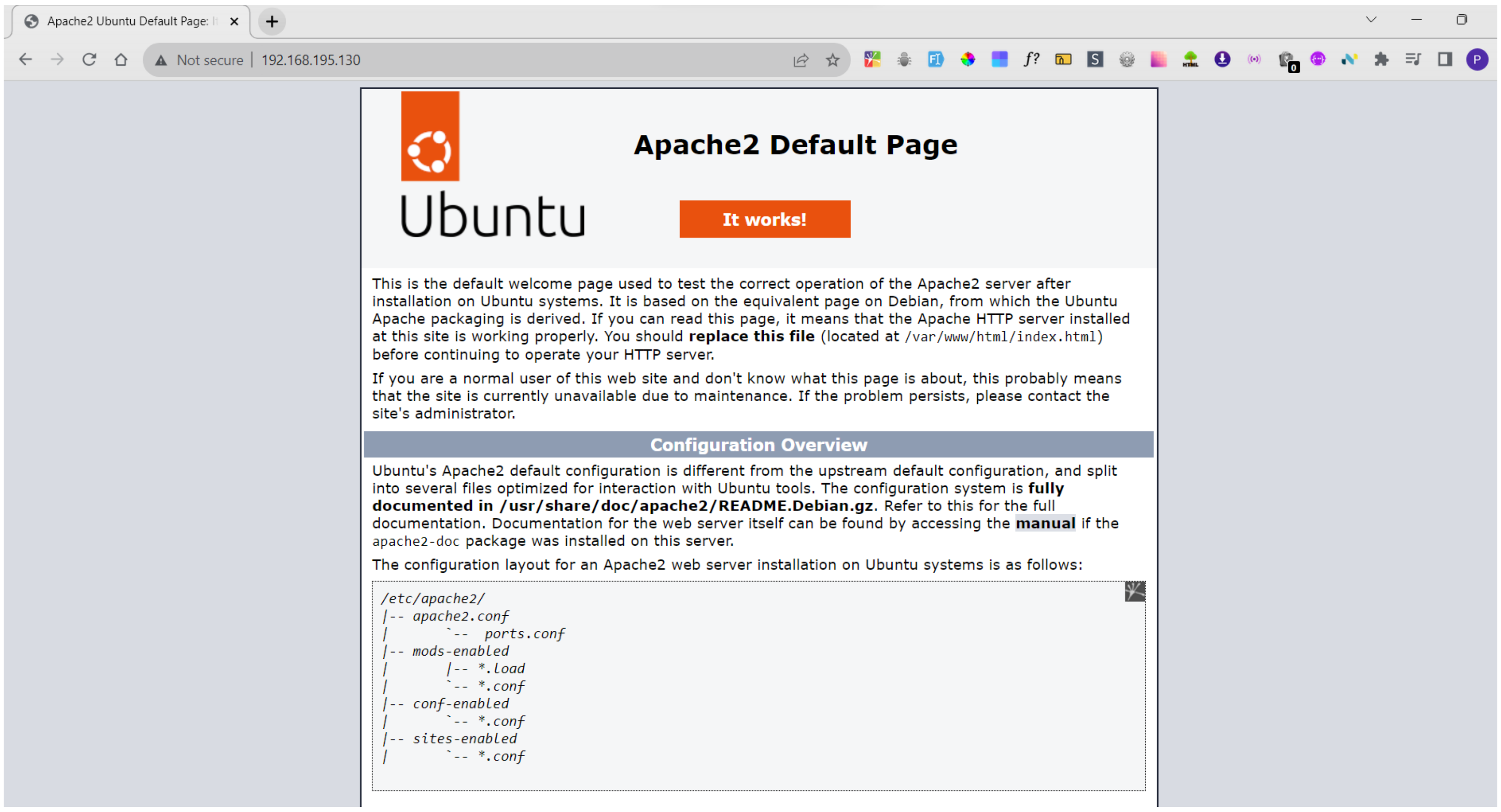
```
iit@lokeshmanikanta:~$ sudo ufw allow 80
[sudo] password for iit:
Rule added
Rule added (v6)
```

sudo ufw status – to check which ports are in open and denied



```
iit@lokeshmanikanta:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
717                        DENY        Anywhere
712                        ALLOW       Anywhere
80                         ALLOW       Anywhere
717 (v6)                   DENY        Anywhere (v6)
712 (v6)                   ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
```

As of port 80 is allowed by firewall so that it loads in another system in the same network



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

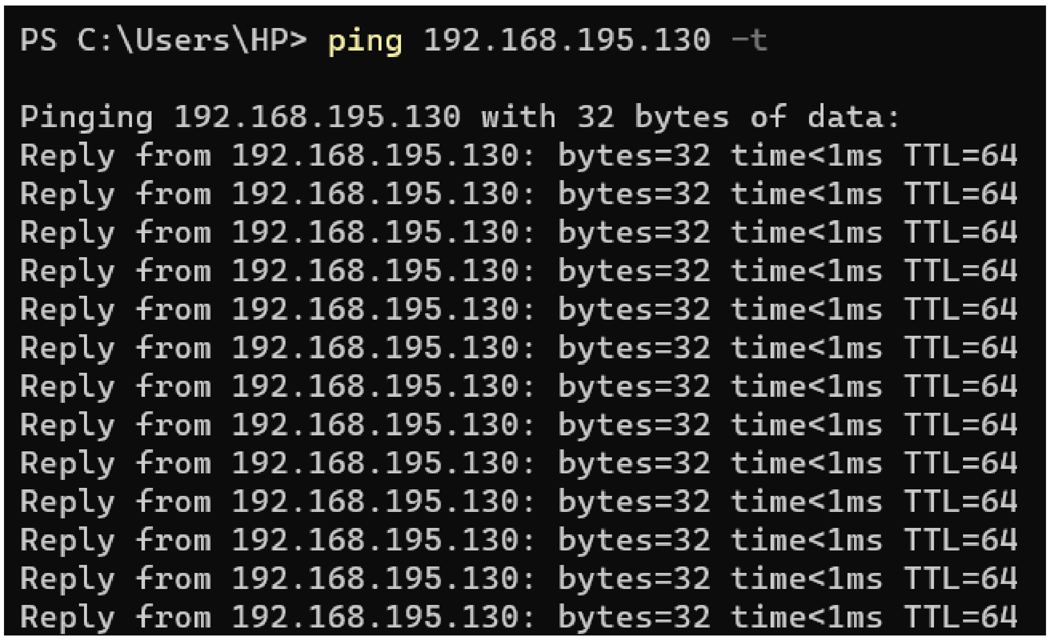
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

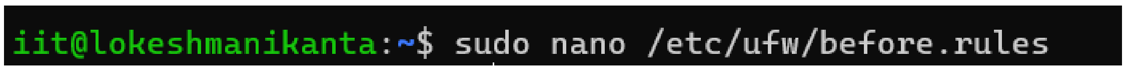
```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

How to stay hidden



```
PS C:\Users\HP> ping 192.168.195.130 -t

Pinging 192.168.195.130 with 32 bytes of data:
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
Reply from 192.168.195.130: bytes=32 time<1ms TTL=64
```

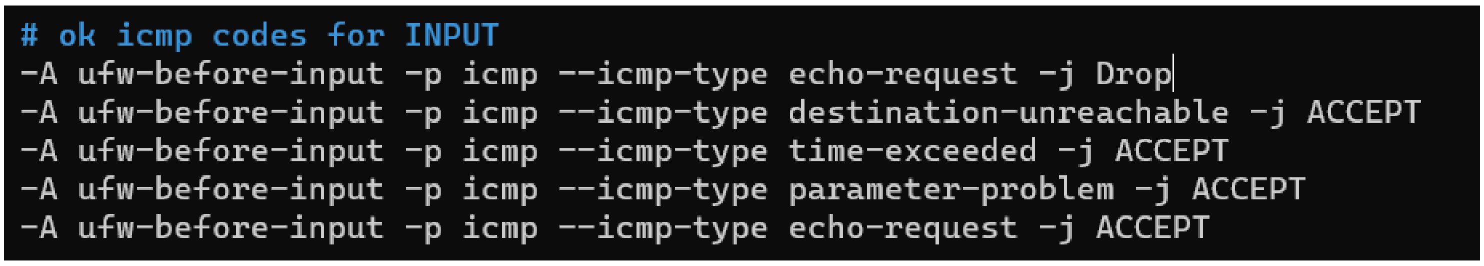
sudo nano /etc/ufw/before.rules



```
iit@lokeshmanikanta:~$ sudo nano /etc/ufw/before.rules
```

Add line in the icmp codes for Input

-A ufw-before-input -p icmp --icmp-type echo-request -j Drop



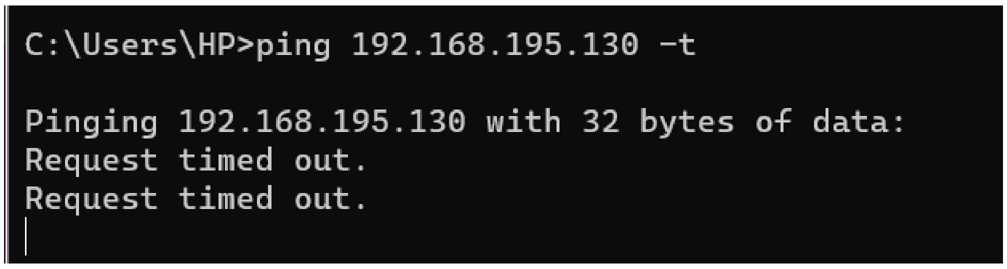
```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type echo-request -j Drop
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT
```

Save the file

Then reboot the system

sudo reboot

now yours linux is hidden

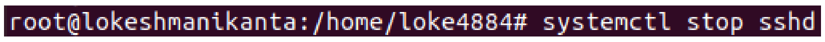


```
C:\Users\HP>ping 192.168.195.130 -t

Pinging 192.168.195.130 with 32 bytes of data:
Request timed out.
Request timed out.
```

Disabling SSH (this is not a good practice)

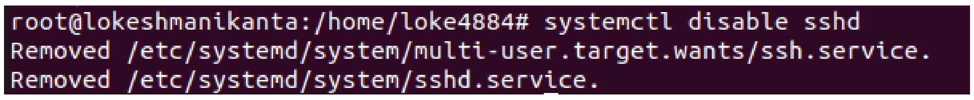
This is most effective such that nobody cant access yours server via ssh

systemctl stop sshd



```
root@lokeshmanikanta:/home/loke4884# systemctl stop sshd
```

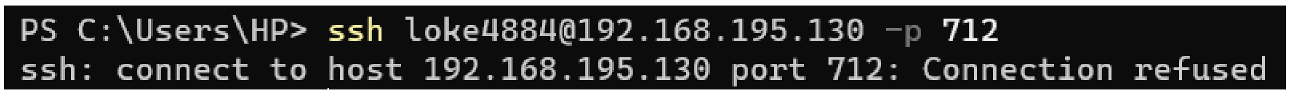
systemctl disable sshd



```
root@lokeshmanikanta:/home/loke4884# systemctl disable sshd
Removed /etc/systemd/system/multi-user.target.wants/ssh.service.
Removed /etc/systemd/system/sshd.service.
```

This is not a good practice because we also cant able to access

This is good if you have no intention use ssh



```
PS C:\Users\HP> ssh loke4884@192.168.195.130 -p 712
ssh: connect to host 192.168.195.130 port 712: Connection refused
```

To recover from this

sudo systemctl enable sshd

sudo systemctl start sshd

systemctl status sshd

Ensure that only specific ip can access ssh

No other machine can access ssh even knowing the port only particular machine which matches the ip can access it

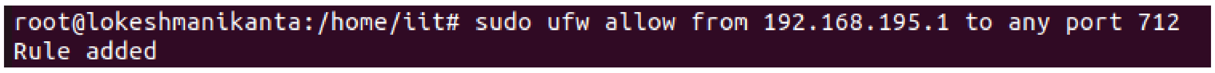
This is more effective if that server/machine has static ip otherwise you have to update firewall rule every time for the server/machine ip

Enable firewall

sudo ufw enable

writing a firewall rule such that ssh can be accessed by only certain ip

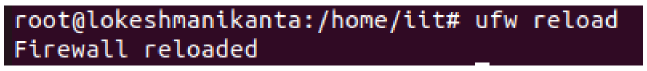
sudo ufw allow from 192.168.195.1 to any port 712



```
root@lokeshmanikanta:/home/iit# sudo ufw allow from 192.168.195.1 to any port 712
Rule added
```

To save the changes in the firewall

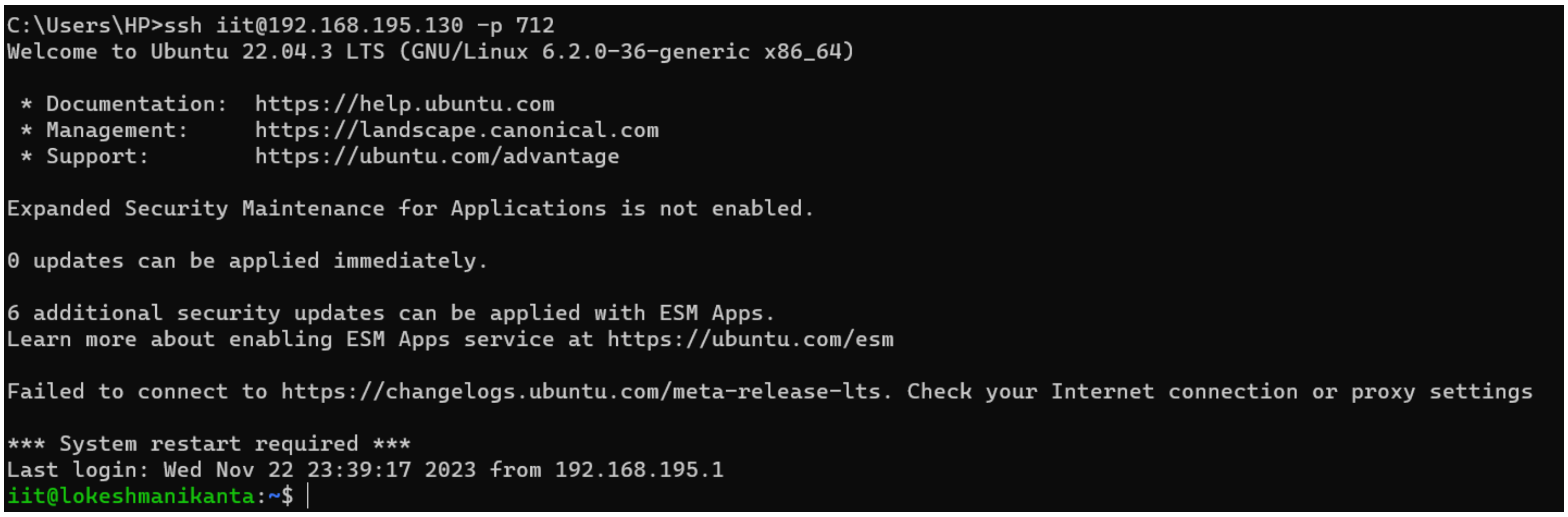
ufw reload



```
root@lokeshmanikanta:/home/iit# ufw reload
Firewall reloaded
```

Open powershell

ssh username@ip -p PortNumber



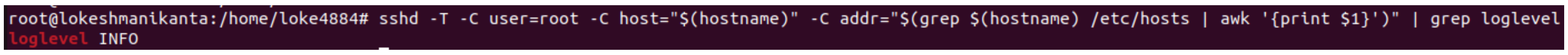
```
C:\Users\HP>ssh iit@192.168.195.130 -p 712
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

*** System restart required ***
Last login: Wed Nov 22 23:39:17 2023 from 192.168.195.1
iit@lokeshmanikanta:~$ 
```

Ensure SSH LogLevel is appropriate (Automated)

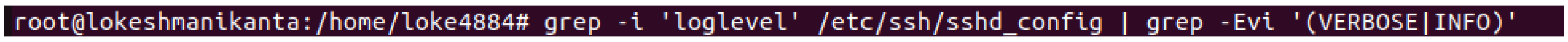
To check LogLevel

Info → it records login activity of SSH, logout activity is eliminated for those users who are disconnected

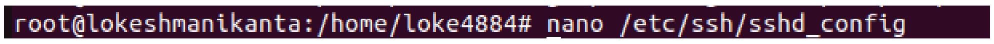
Verbose → Verbose logging is a computer logging method that records more information than the standard logging process,It records login and logout activities , this is important for ssh key management



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep loglevel
loglevel INFO
```

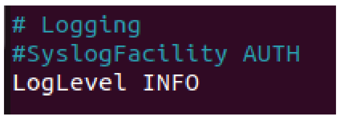


```
root@lokeshmanikanta:/home/loke4884# grep -i 'loglevel' /etc/ssh/sshd_config | grep -Evi '(VERBOSE|INFO)'
```



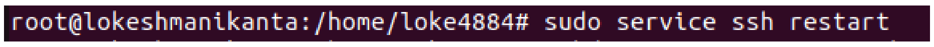
```
root@lokeshmanikanta:/home/loke4884# nano /etc/ssh/sshd_config
```

Uncommit LogLevel INFO



```
# Logging
#SyslogFacility AUTH
LogLevel INFO
```

Save the file and restart the SSH service for the changes to take effect:



```
root@lokeshmanikanta:/home/loke4884# sudo service ssh restart
```

the newer OpenSSH releases do not need a verbose mode setting anymore as the required SSH key activity information is written into the syslog by the default OpenSSH config.

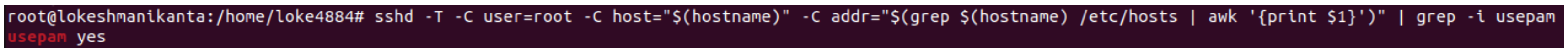
Ensure SSH PAM is enabled (Automated)

sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i usepam

usepam" in the output, it means that PAM is being used in the SSH configuration.

If set to yes this will enable PAM authentication using ChallengeResponseAuthentication and PasswordAuthentication directives in addition to PAM account and session module processing for all authentication types.

Usepam yes → this ensures if you want to restrict access to services based off of IP, time or other factors of the account



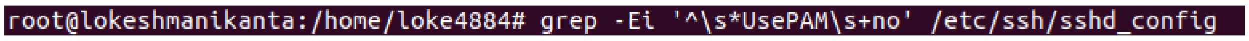
```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i usepam
usepam yes
```

grep -Ei '^\s*UsePAM\s+no' /etc/ssh/sshd_config

This command should not return anything

This is typically used to check if the SSH server is configured to disable the use of PAM (Pluggable Authentication Modules).

UsePAM is set to "no" in the SSH server configuration file. This could mean that the system is configured to not use PAM for authentication in the SSH server.



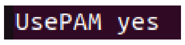
```
root@lokeshmanikanta:/home/loke4884# grep -Ei '^\s*UsePAM\s+no' /etc/ssh/sshd_config
```

If the SSH server is configured to disable the use of PAM then

Go to configuration file of ssh

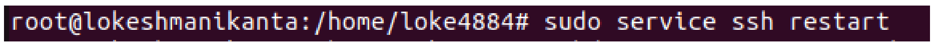
nano /etc/ssh/sshd_config

enable UsePAM



```
UsePAM yes
```

Save the file and restart the SSH service for the changes to take effect:

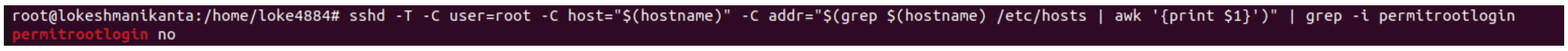


```
root@lokeshmanikanta:/home/loke4884# sudo service ssh restart
```

Ensure SSH root login is disabled (Automated)

sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i permitrootlogin

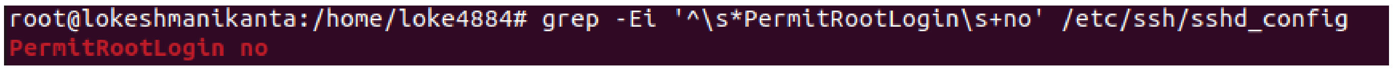
This command is checking specific configuration settings of the SSH daemon without starting it. It verifies whether the configuration allows the root user, connections from the current host, and connections from the IP address associated with the current host name. Finally, it checks if the configuration includes the option PermitRootLogin, indicating whether root logins are permitted.



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i permitrootlogin
permitrootlogin no
```

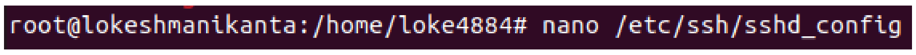
OR

grep -Ei '^\s*PermitRootLogin\s+no' /etc/ssh/sshd_config



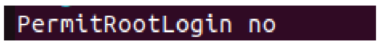
```
root@lokeshmanikanta:/home/loke4884# grep -Ei '^\s*PermitRootLogin\s+no' /etc/ssh/sshd_config
PermitRootLogin no
```

If output of that command doesn’t show PermitRootLogin no then go to ssh_config file and configure manually set PermitRootLogin to no



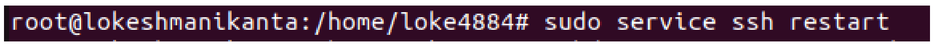
```
root@lokeshmanikanta:/home/loke4884# nano /etc/ssh/sshd_config
```

Set PermitRootLogin to no , by default PermitRootLogin is set to prohibit-password



```
PermitRootLogin no
```

Save the file and restart the SSH service for the changes to take effect:



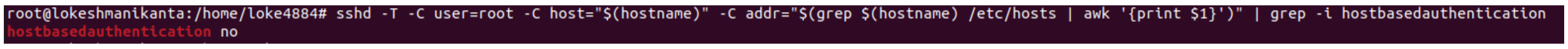
```
root@lokeshmanikanta:/home/loke4884# sudo service ssh restart
```

Ensure SSH HostbasedAuthentication is disabled

Host-based authentication allows users to log in based on the host they are connecting from, rather than using traditional password or key-based authentication,so set hostbasedAuthentication should set to no

sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -I hostbasedauthentication

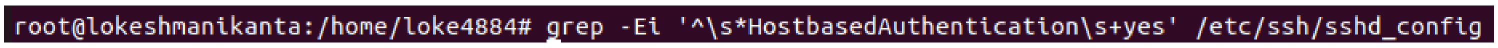
indicate whether the SSH daemon is configured to allow host-based authentication.



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i hostbasedauthentication
hostbasedauthentication no
```

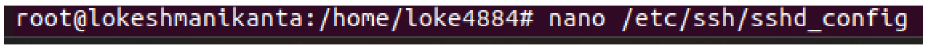
grep -Ei '^\s*HostbasedAuthentication\s+yes' /etc/ssh/sshd_config

this command should return nothing



```
root@lokeshmanikanta:/home/loke4884# grep -Ei '^\s*HostbasedAuthentication\s+yes' /etc/ssh/sshd_config
```

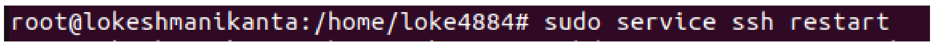
If hostbasedauthentication is not set to no then go to sshd config file



```
root@lokeshmanikanta:/home/loke4884# nano /etc/ssh/sshd_config
```

Set hostbasedauthentication to no or remove hostbasedauthentication

Save the file and restart the SSH service for the changes to take effect:



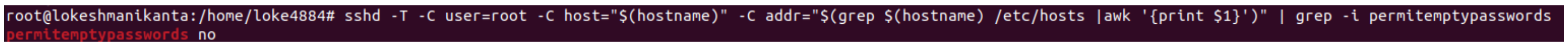
```
root@lokeshmanikanta:/home/loke4884# sudo service ssh restart
```

Ensure SSH PermitEmptyPasswords is disabled

The PermitEmptyPasswords parameter specifies if the SSH server allows login to accounts with empty password strings

Make sure permitemptypasswords no

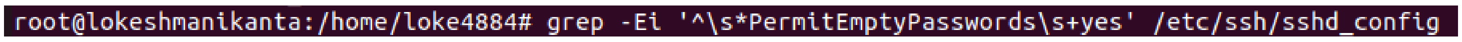
sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |awk '{print $1}')" | grep -i permitemptypasswords



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |awk '{print $1}')" | grep -i permitemptypasswords
permitemptypasswords no
```

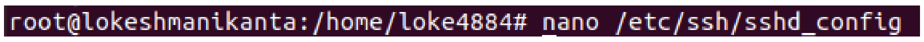
grep -Ei '^\s*PermitEmptyPasswords\s+yes' /etc/ssh/sshd_config

nothing should return



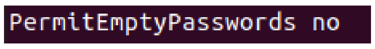
```
root@lokeshmanikanta:/home/loke4884# grep -Ei '^\s*PermitEmptyPasswords\s+yes' /etc/ssh/sshd_config
```

To set permitemptypasswords to no



```
root@lokeshmanikanta:/home/loke4884# nano /etc/ssh/sshd_config
```

Enable PermitEmptyPasswords no

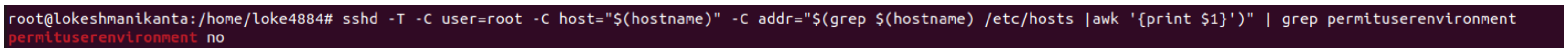


```
PermitEmptyPasswords no
```

Ensure SSH PermitUserEnvironment is disabled

Permitting users the ability to set environment variables through the SSH daemon could potentially allow users to bypass security controls (e.g. setting an execution path that has SSH executing trojan's programs)

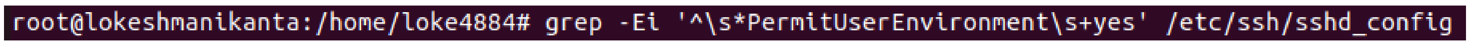
sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |awk '{print $1}')" | grep permituserenvironment



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts |awk '{print $1}')" | grep permituserenvironment
permituserenvironment no
```

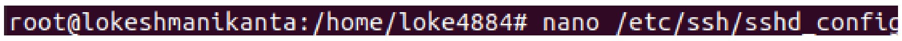
grep -Ei '^\s*PermitUserEnvironment\s+yes' /etc/ssh/sshd_config

should return nothing



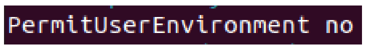
```
root@lokeshmanikanta:/home/loke4884# grep -Ei '^\s*PermitUserEnvironment\s+yes' /etc/ssh/sshd_config
```

Go to sshd_config



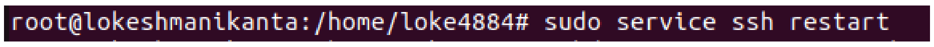
```
root@lokeshmanikanta:/home/loke4884# nano /etc/ssh/sshd_config
```

set PermitUserEnvironment no



```
PermitUserEnvironment no
```

Save the file and restart the SSH service for the changes to take effect:

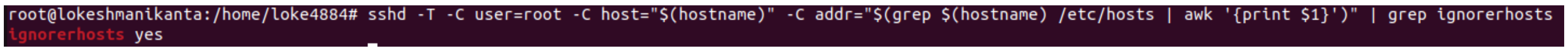


```
root@lokeshmanikanta:/home/loke4884# sudo service ssh restart
```

Ensure SSH IgnoreRhosts is enabled (Automated)

If you want to ignore .rhosts and .shosts files in SSH, you typically set the IgnoreRhosts option in your SSH configuration

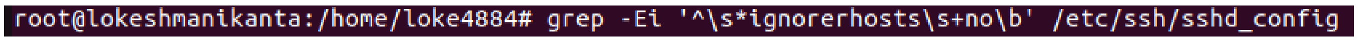
sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep ignorerhosts



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep ignorerhosts
ignorerhosts yes
```

No output should be return

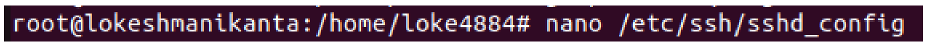
grep -Ei '^\s*ignorerhosts\s+no\b' /etc/ssh/sshd_config



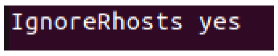
```
root@lokeshmanikanta:/home/loke4884# grep -Ei '^\s*ignorerhosts\s+no\b' /etc/ssh/sshd_config
```

To enable Ignorehosts go to sshd configuration file

nano /etc/ssh/sshd_config

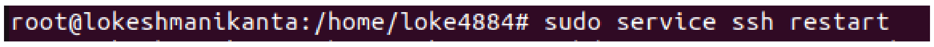


```
root@lokeshmanikanta:/home/loke4884# nano /etc/ssh/sshd_config
```



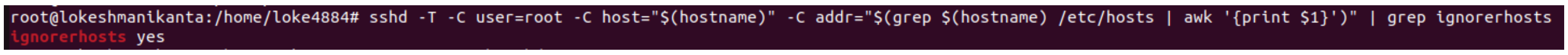
```
IgnoreRhosts yes
```

Save the file and restart the SSH service for the changes to take effect:



```
root@lokeshmanikanta:/home/loke4884# sudo service ssh restart
```

Run the following command to check whether ignorerhosts enabled



```
root@lokeshmanikanta:/home/loke4884# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep ignorerhosts
ignorerhosts yes
```

Ensure SSH X11 forwarding is disabled (Automated)

Configuration of XLaunch

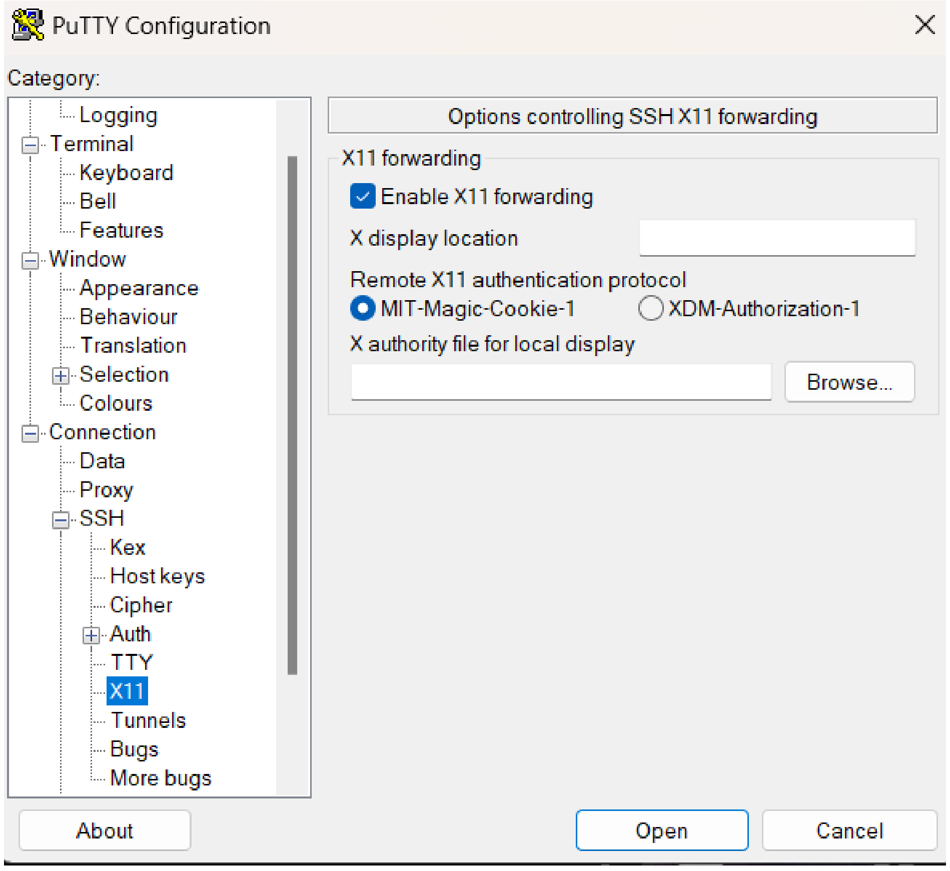
Multiple windows → Start no client → enable clipboard(if you desired) then click on Next → Finish

Now we are having X server running on ours local windows machine

Open Putty

Go to Connection → SSH → Auth → X11

Make sure to Enable X11 forwarding and MIT-Magic-Cookie-1 is checked



PuTTY Configuration

Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

X display location

Remote X11 authentication protocol

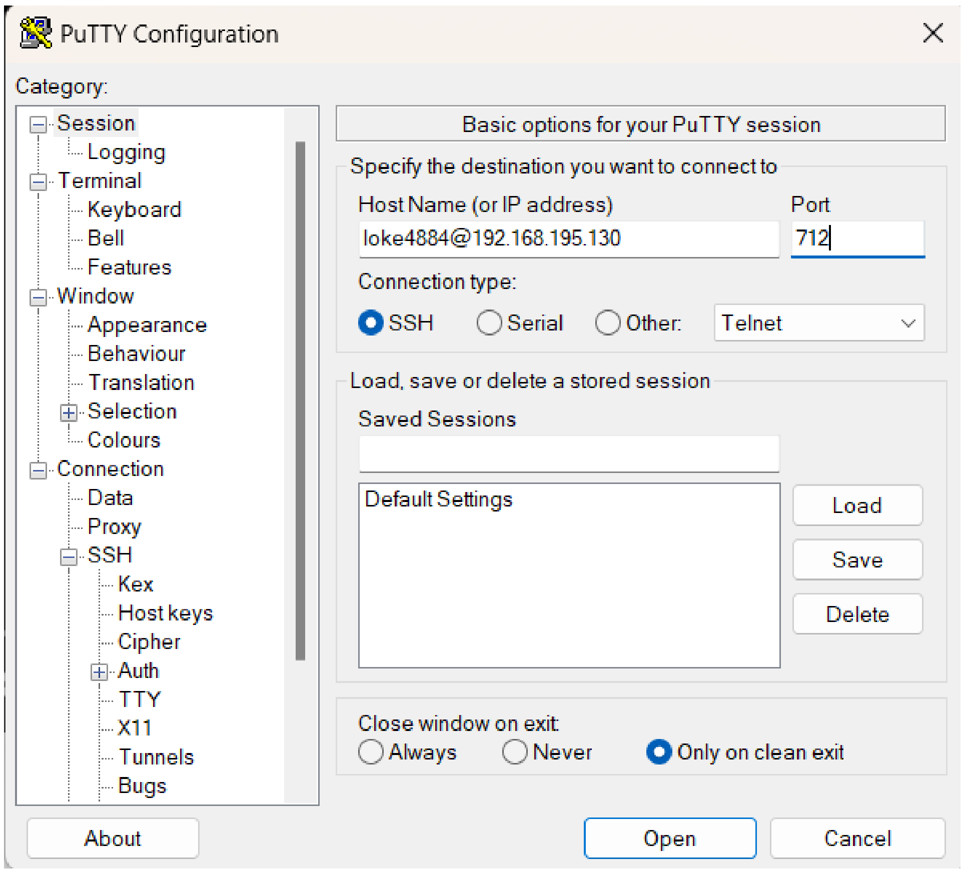
● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display

Go to Session → choose SSH

username@IpOfServer

and give port (which is configured to ssh)



Host Name (or IP address): loke4884@192.168.195.130

Port: 712

Connection type: ● SSH ○ Serial ○ Other: Telnet

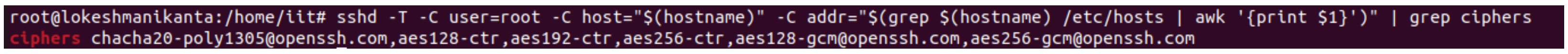
Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Ensure only strong Ciphers are used (Automated)

sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep ciphers

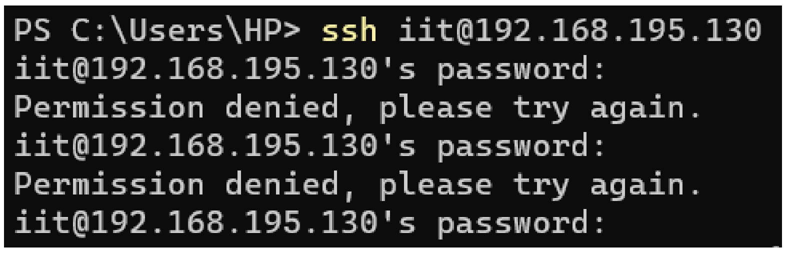
output should not contain any of weak ciphers



```
root@lokeshmanikanta:/home/iit# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep ciphers
ciphers chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com
```

Ensure SSH MaxAuthTries is set to 4 or less (Automated)

The MaxAuthTries parameter specifies the maximum number of authentication attempts permitted per connection. When the login failure count reaches half the number, error messages will be written to the syslog file detailing the login failure.

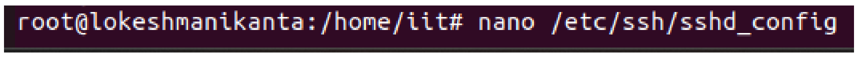


```
PS C:\Users\HP> ssh iit@192.168.195.130
iit@192.168.195.130's password:
Permission denied, please try again.
iit@192.168.195.130's password:
Permission denied, please try again.
iit@192.168.195.130's password:
```

Go to ssh configuration file

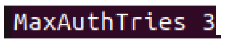
Set MaxAuthTries as how much you wanted

Keep as low attempts as possible to avoid brute force to get access through ssh

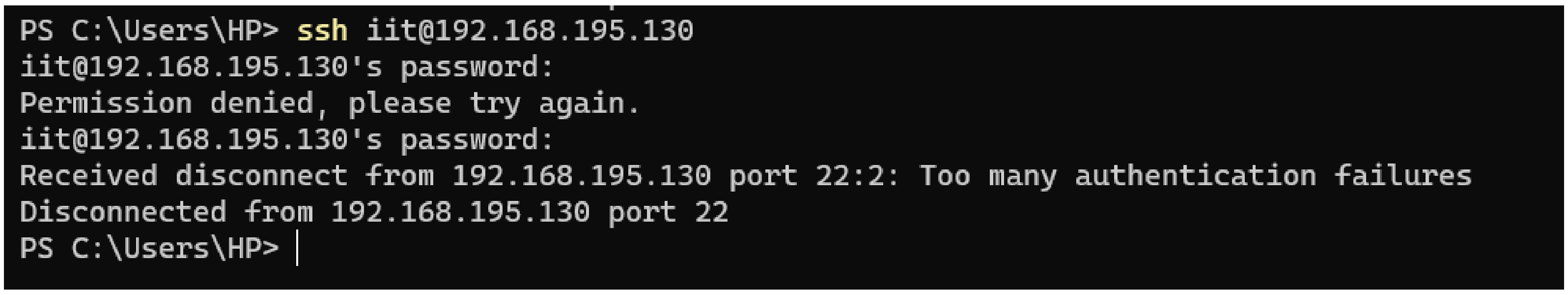


```
root@lokeshmanikanta:/home/iit# nano /etc/ssh/sshd_config
```

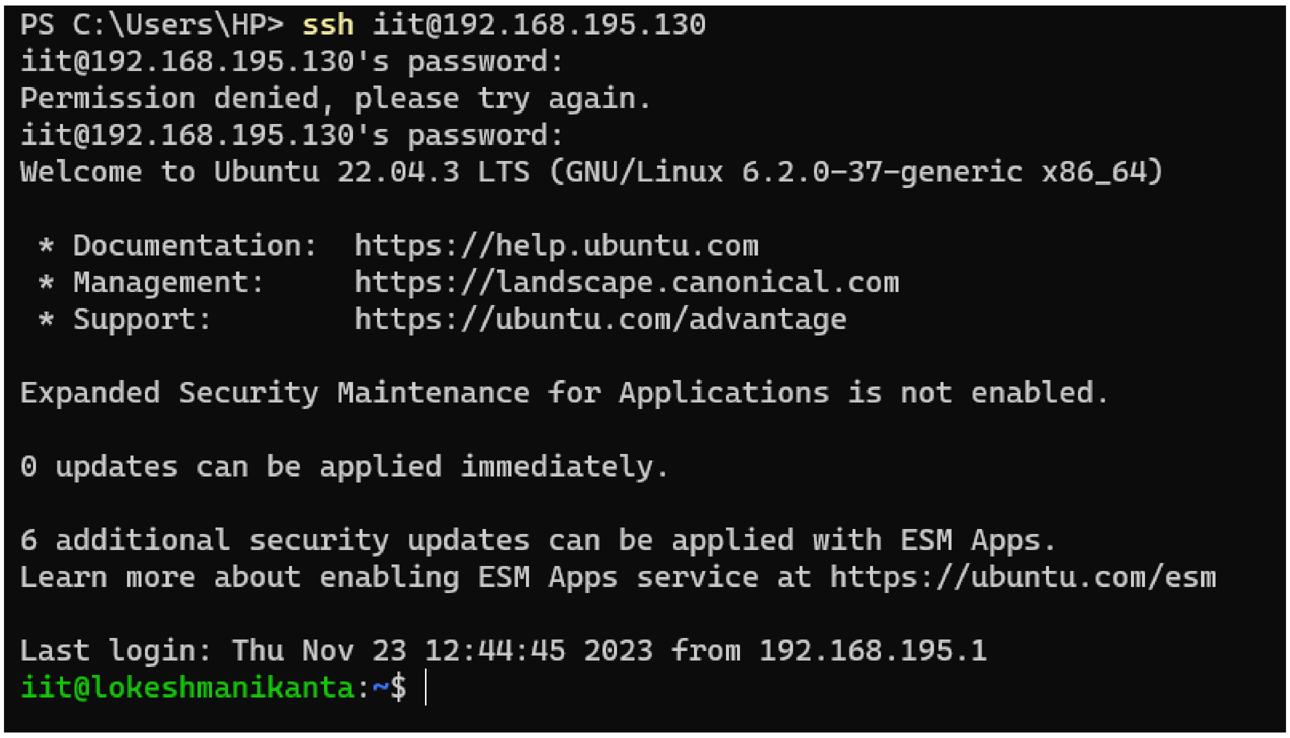
In the 2nd attempt to access with wrong password it doesnot allow to login



```
MaxAuthTries 3
```

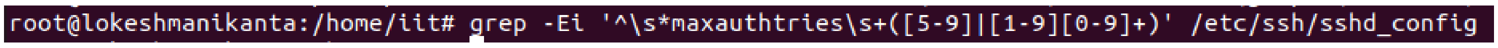


```
PS C:\Users\HP> ssh iit@192.168.195.130
iit@192.168.195.130's password:
Permission denied, please try again.
iit@192.168.195.130's password:
Received disconnect from 192.168.195.130 port 22:2: Too many authentication failures
Disconnected from 192.168.195.130 port 22
PS C:\Users\HP>
```



```
PS C:\Users\HP> ssh iit@192.168.195.130
iit@192.168.195.130's password:
Permission denied, please try again.
iit@192.168.195.130's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-37-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

6 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Thu Nov 23 12:44:45 2023 from 192.168.195.1
iit@lokeshmanikanta:~$
```

Make sure to run this command such that no output to get(to check enabled or not for MaxAuthorities)
grep -Ei '^\s*maxauthtries\s+([5-9]||[1-9][0-9]+)' /etc/ssh/sshd_config



```
root@lokeshmanikanta:/home/iit# grep -Ei '^\s*maxauthtries\s+([5-9]||[1-9][0-9]+)' /etc/ssh/sshd_config
```

Ensure SSH MaxStartups is configured (Automated)

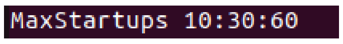
The MaxStartups configuration option in SSH (Secure Shell) is used to limit the number of concurrent unauthenticated connections to the SSH server. This option can help prevent resource exhaustion caused by a large number of incomplete or failed authentication attempts.

To protect a system from denial of service due to a large number of pending authentication connection attempts, use the rate limiting function of MaxStartups to protect availability of sshd logins and prevent overwhelming the daemon

Go to configuration file of ssh

nano /etc/ssh/sshd_config

enable MaxStartups



```
MaxStartups 10:30:60
```

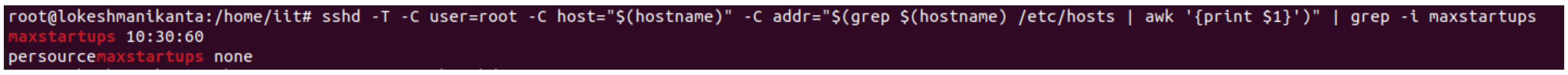
sets limits on the number of unauthenticated connections. The server will allow all connections if there are fewer than 10. If there are between 10 and 30 connections, the server will randomly drop connections until the count reaches 10. If there are more than 30 connections, the server will randomly drop connections until the count reaches 30.

After applying changes in ssh configuration

systemctl restart ssh

To check that changes enabled

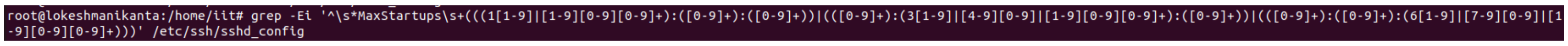
sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i maxstartups



```
root@lokeshmanikanta:/home/iit# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i maxstartups
maxstartups 10:30:60
persourcemaxstartups none
```

or

grep -Ei '^\s*MaxStartups\s+(((1[1-9]|[1-9][0-9][0-9]+):([0-9]+):([0-9]+))|(([0-9]+):(3[1-9]|[4-9][0-9]|[1-9][0-9][0-9]+):([0-9]+))|(([0-9]+):([0-9]+):(6[1-9]|[7-9][0-9]|[1-9][0-9][0-9]+)))' /etc/ssh/sshd_config



```
root@lokeshmanikanta:/home/iit# grep -Ei '^\s*MaxStartups\s+(((1[1-9]|[1-9][0-9][0-9]+):([0-9]+):([0-9]+))|(([0-9]+):(3[1-9]|[4-9][0-9]|[1-9][0-9][0-9]+):([0-9]+))|(([0-9]+):([0-9]+):(6[1-9]|[7-9][0-9]|[1
-9][0-9][0-9]+)))' /etc/ssh/sshd_config
```

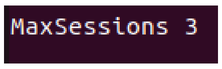
Ensure SSH MaxSessions is set to 10 or less (Automated)

To protect a system from denial of service due to a large number of concurrent sessions, use the rate limiting function of MaxSessions to protect availability of sshd logins and prevent overwhelming the daemon.

Go to conguartion of ssh

nano /etc/ssh/sshd_config

```
root@lokeshmanikanta:/home/iit# nano /etc/ssh/sshd_config
```



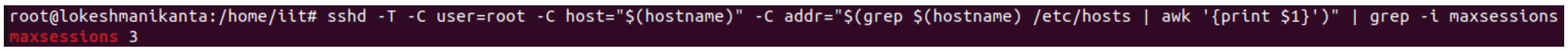
```
MaxSessions 3
```

Restart ssh to appear changes

```
root@lokeshmanikanta:/home/iit# systemctl restart ssh
```

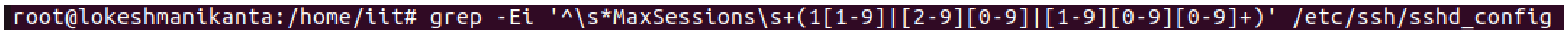
Check maxsessions is set or not

sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i maxsessions



```
root@lokeshmanikanta:/home/iit# sshd -T -C user=root -C host="$(hostname)" -C addr="$(grep $(hostname) /etc/hosts | awk '{print $1}')" | grep -i maxsessions
maxsessions 3
```

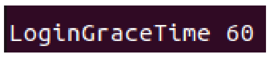
grep -Ei '^\s*MaxSessions\s+(1[1-9]|[2-9][0-9]|[1-9][0-9][0-9]+)' /etc/ssh/sshd_config



```
root@lokeshmanikanta:/home/iit# grep -Ei '^\s*MaxSessions\s+(1[1-9]|[2-9][0-9]|[1-9][0-9][0-9]+)' /etc/ssh/sshd_config
```

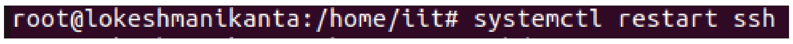
Ensure SSH LoginGraceTime is set to one minute or less

nano /etc/ssh/sshd_config



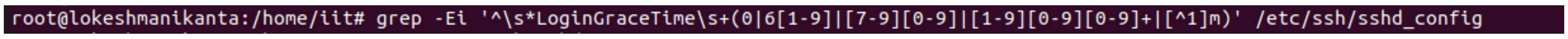
```
LoginGraceTime 60
```

Restart ssh



```
root@lokeshmanikanta:/home/iit# systemctl restart ssh
```

Nothing should return



```
root@lokeshmanikanta:/home/iit# grep -Ei '^\s*LoginGraceTime\s+(0|6[1-9]|[7-9][0-9]|[1-9][0-9][0-9]+|[^1]m)' /etc/ssh/sshd_config
```