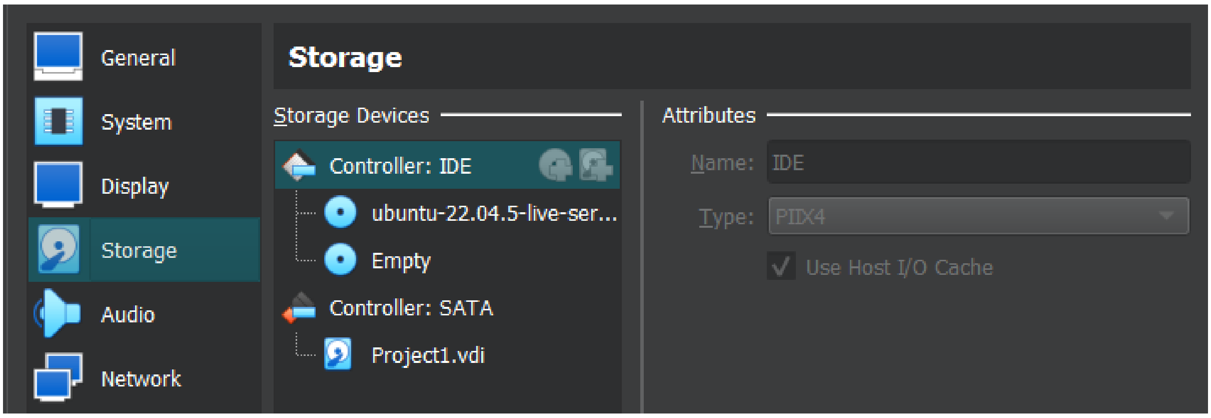
First Project

I set up a cloud-based server using Ubuntu Live Server in a virtual box, naming it 'FirstProject' :



Storage

Storage Devices

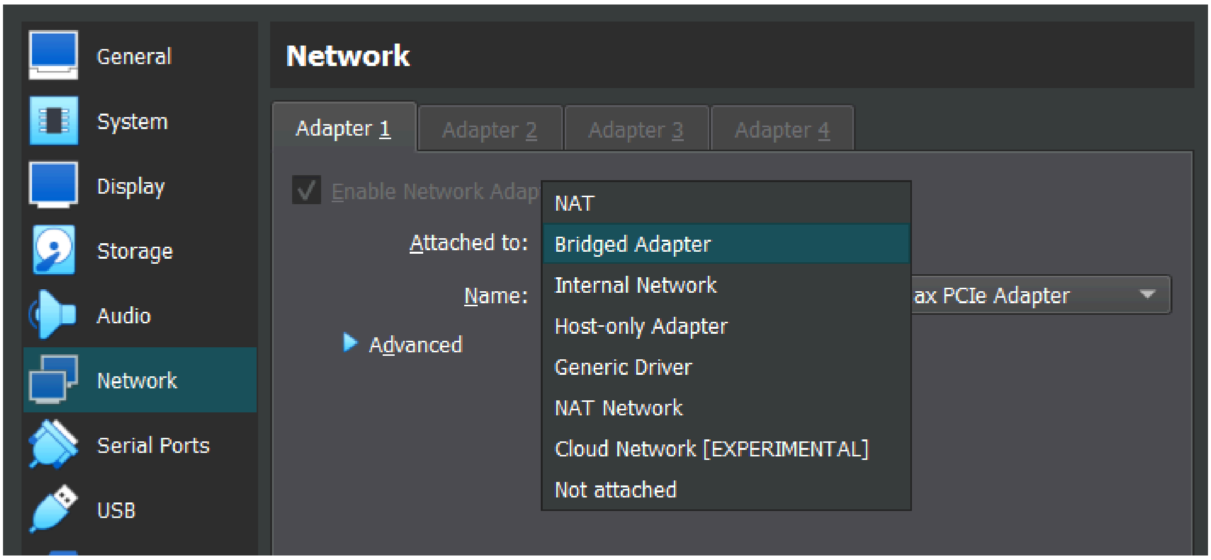
- Controller: IDE
  - ubuntu-22.04.5-live-ser...
  - Empty
- Controller: SATA
  - Project1.vdi

Attributes

Name: IDE

Type: PIIX4

☑ Use Host I/O Cache

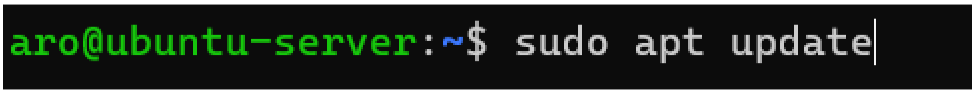


Network

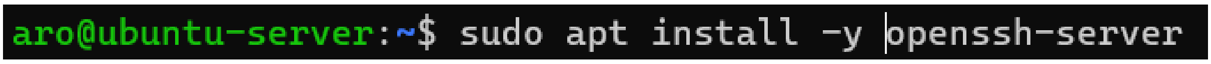
☑ Enable Network Adap

Attached to: Bridged Adapter

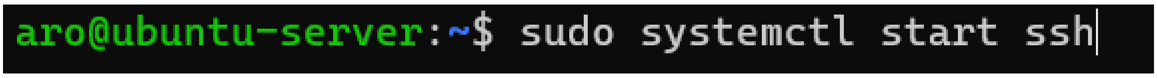
Name: ax PCIe Adapter



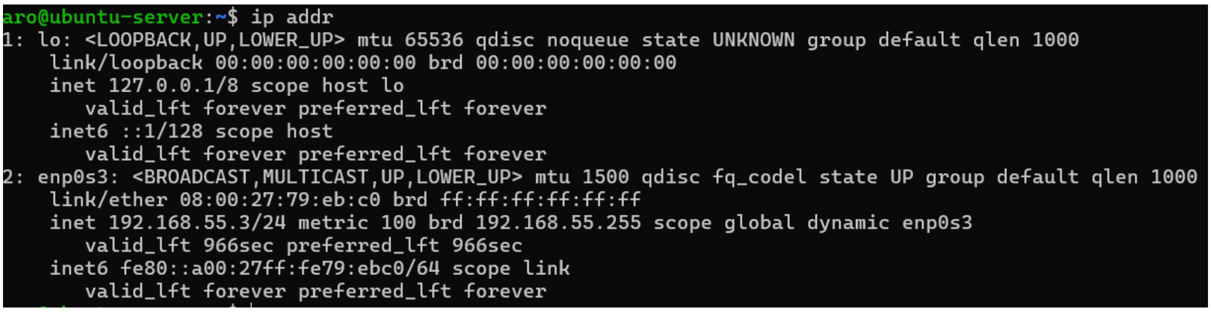
```
aro@ubuntu-server:~$ sudo apt update
```



```
aro@ubuntu-server:~$ sudo apt install -y openssh-server
```

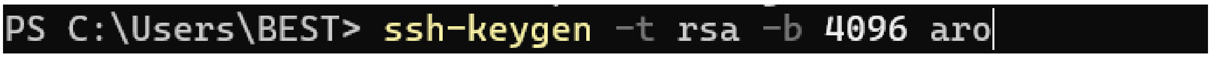


```
aro@ubuntu-server:~$ sudo systemctl start ssh
```

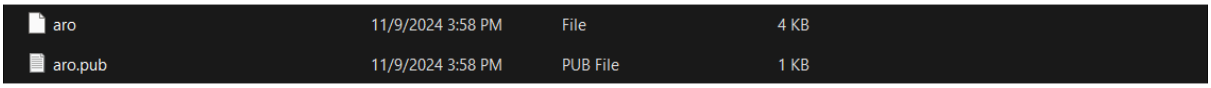


```
aro@ubuntu-server:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:79:eb:c0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.55.3/24 metric 100 brd 192.168.55.255 scope global dynamic enp0s3
       valid_lft 966sec preferred_lft 966sec
    inet6 fe80::a00:27ff:fe79:ebc0/64 scope link
       valid_lft forever preferred_lft forever
```

We have generated a new key named 'aro'.

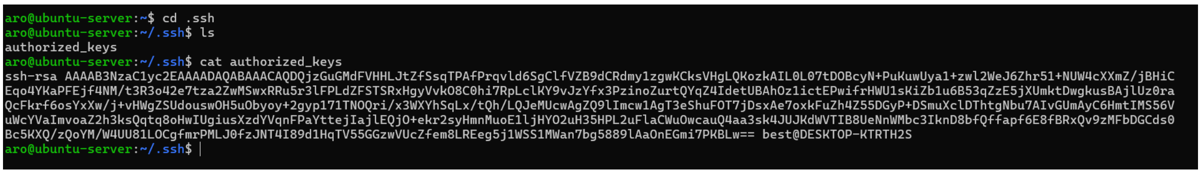


```
PS C:\Users\BEST> ssh-keygen -t rsa -b 4096 aro
```



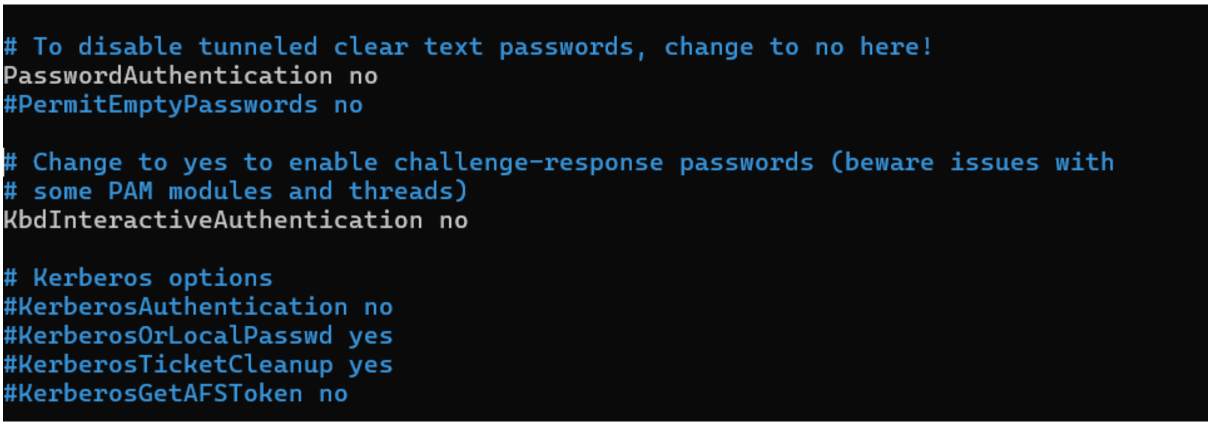
```
aro        11/9/2024 3:58 PM    File        4 KB
aro.pub    11/9/2024 3:58 PM    PUB File    1 KB
```

We created a new user named 'aro,' granted root privileges, and then opened the user’s directory to add our private key to the authorized_keys file.



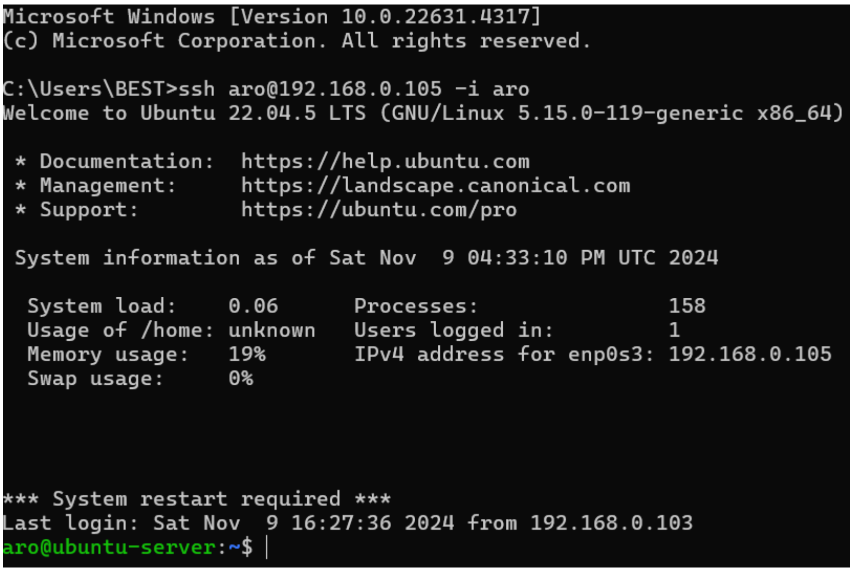
```
aro@ubuntu-server:~$ cd .ssh
aro@ubuntu-server:~/.ssh$ ls
authorized_keys
aro@ubuntu-server:~/.ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDQjzGuGMdFVHHLJtZfSsqTPAfPrqvld6SgClfVZB9dCRdmy1zgwKCksVHgLQKozkAIL0L07tDOBcyN+PuKuwUya1+zwl2WeJ6Zhr51+NUW4cXXmZ/jBHiC
Eqo4YKaPFEjf4NM/t3R3o42e7tza2ZwMSwxRRu5r3lFPLdZFSTSRxHgyVvkO8C0hi7RpLclKY9vJzYfx3PzinoZurtQYqZ4IdetUBAhOz1ictEPwifrHWU1sKiZb1u6B53qZzE5jXUmktDwgkusBAjlUz0ra
QcFkrf6osYxXw/j+vHWgZSUdouswOH5uObyoy+2gyp171TNOQri/x3WXYhSqLx/tQh/LQJeMUcwAgZQ9lImcw1AgT3eShuFOT7jDsxAe7oxkFuZh4Z55DGyP+DSmuXclDThtgNbu7AIvGUmAyC6HmtIMS56V
uWcYVaImvoaZ2h3ksQqtq8oHwIUgiusXzdYVqnFPaYttejIajlEQjO+ekr2syHmnMuoE1ljHYO2uH35HPL2uFlaCWuOwcauQ4aa3sk4JUJKdWVTIB8UeNnWMbc3IknD8bfQffapf6E8fBRxQv9zMFbDGCds0
Bc5KXQ/zQoYM/W4UU81LOCgfmrPMLJ0fzJNT4I89d1HqTV55GGzwVUcZfem8LREeg5j1WSS1MWan7bg5889lAaOnEGmi7PKBLw== best@DESKTOP-KTRTH2S
aro@ubuntu-server:~/.ssh$ |
```

Password authentication must be off.



```
# 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
```

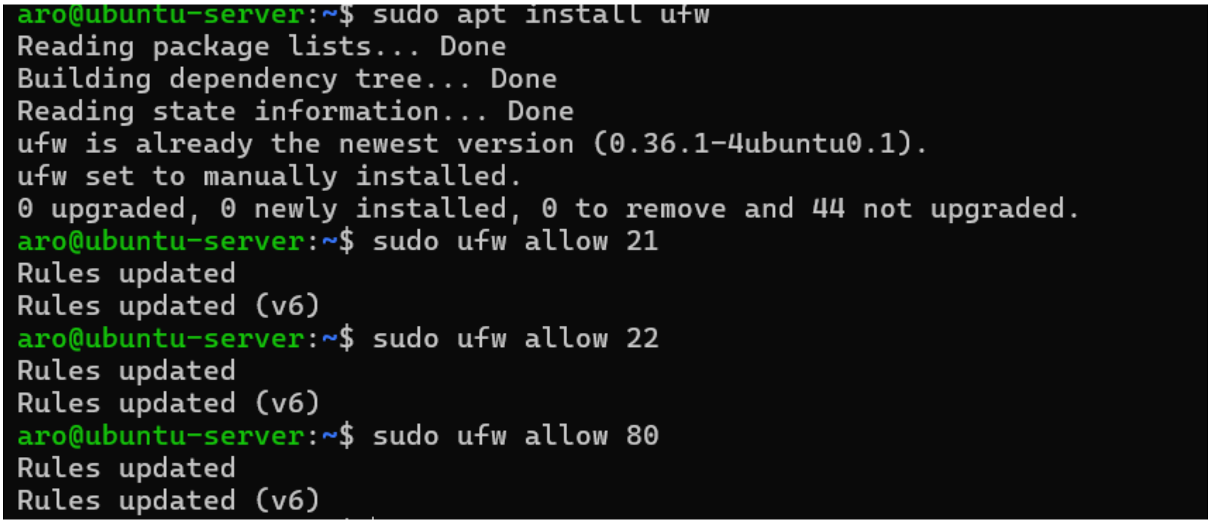
Now, we connect to the server using only the key.



```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\BEST>ssh aro@192.168.0.105 -i aro
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-119-generic x86_64)

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

 System information as of Sat Nov  9 04:33:10 PM UTC 2024

  System load:     0.06      Processes:              158
  Usage of /home: unknown    Users logged in:        1
  Memory usage:    19%       IPv4 address for enp0s3: 192.168.0.105
  Swap usage:      0%



*** System restart required ***
Last login: Sat Nov  9 16:27:36 2024 from 192.168.0.103
aro@ubuntu-server:~$ |
```

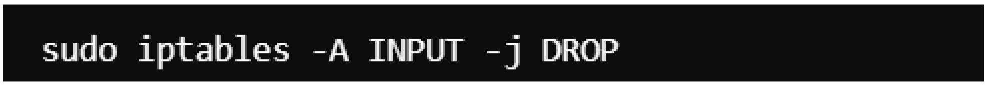
The server is now fully prepared and ready for use.

Now install and configure firewall (ufw,iptables) to allow only port 21, 22 and 80 on the server.



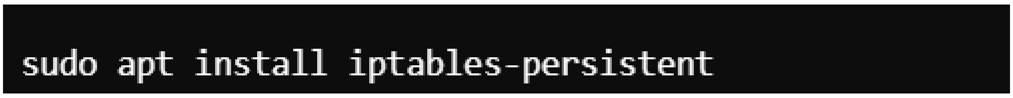
```
aro@ubuntu-server:~$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.1-4ubuntu0.1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 44 not upgraded.
aro@ubuntu-server:~$ sudo ufw allow 21
Rules updated
Rules updated (v6)
aro@ubuntu-server:~$ sudo ufw allow 22
Rules updated
Rules updated (v6)
aro@ubuntu-server:~$ sudo ufw allow 80
Rules updated
Rules updated (v6)
```

Now, we are blocking all incoming network connections that do not match any previously defined rules in iptables. This ensures that only specifically allowed traffic is accepted, while all other connections are silently dropped. This final rule strengthens our server's security by preventing unauthorized access attempts and reducing the risk of attacks.

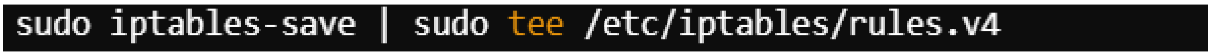


```
sudo iptables -A INPUT -j DROP
```

Next, we save the current configurations to a specific file, which guarantees that our firewall rules remain consistent and are automatically applied every time the server restarts. This process ensures the stability and reliability of our security settings.

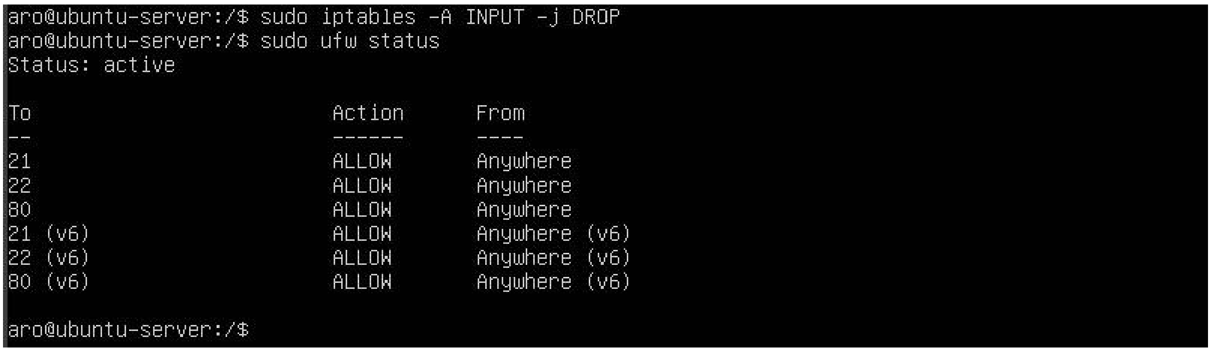


```
sudo apt install iptables-persistent
```



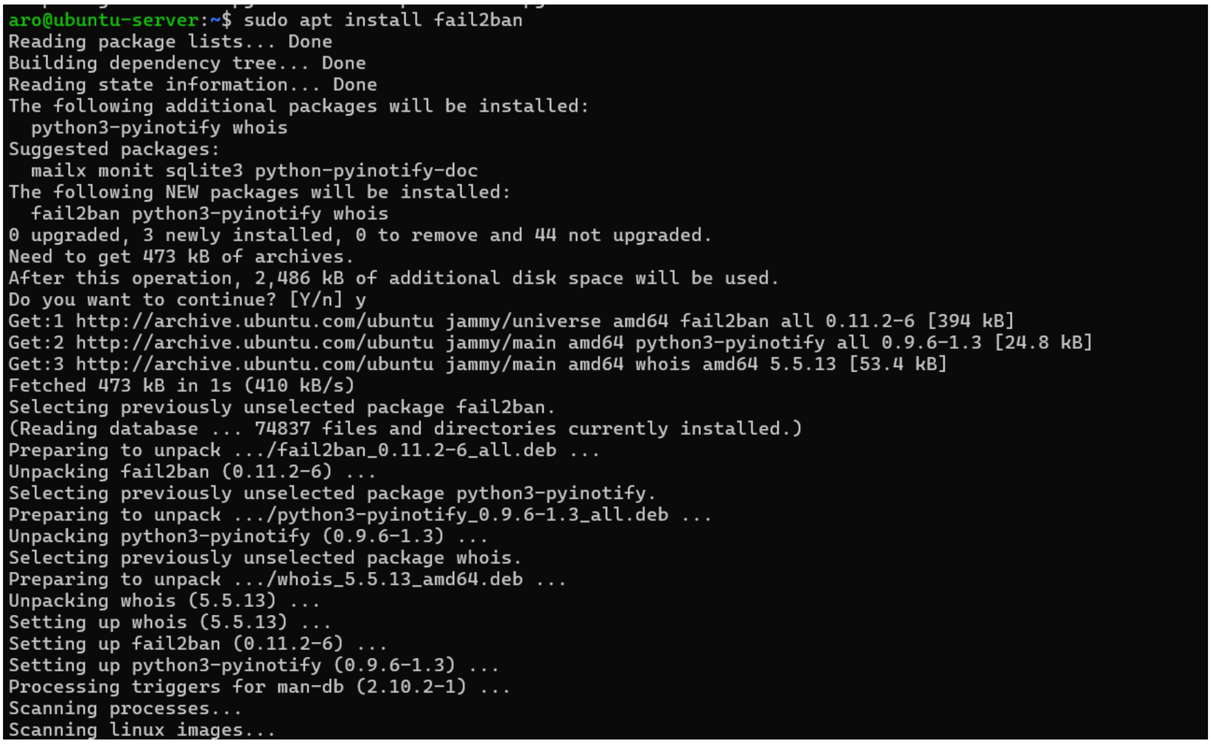
```
sudo iptables-save | sudo tee /etc/iptables/rules.v4
```

Now this command provides a list of the active firewall rules, showing the allowed connections and ensuring that only the specified ports (such as SSH, FTP, and HTTP) are open while all others are blocked, confirming that the firewall is configured correctly and is working as intended.



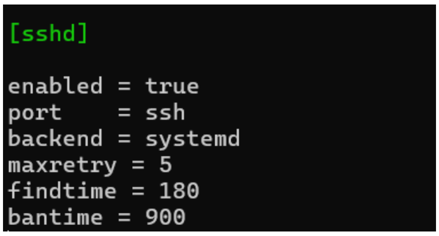
```
aro@ubuntu-server:/$ sudo iptables -A INPUT -j DROP
aro@ubuntu-server:/$ sudo ufw status
Status: active

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

aro@ubuntu-server:/$
```

Now, let's configure Fail2ban to prevent attacks such as login brute force, even if password authentication is off.



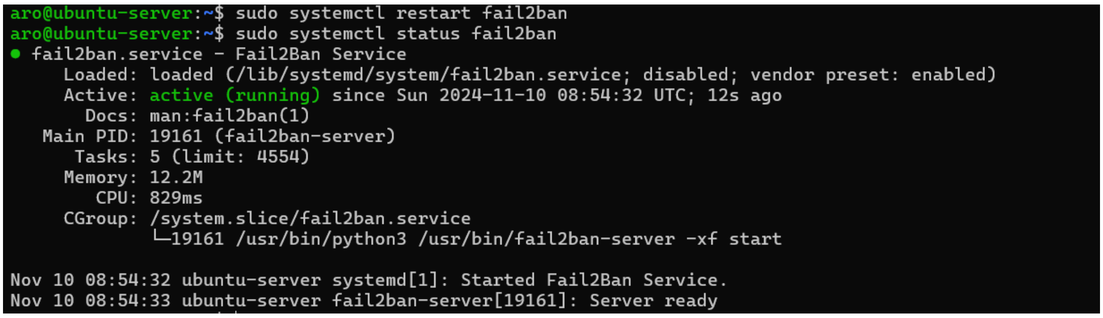
```
aro@ubuntu-server:~$ sudo apt install fail2ban
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 44 not upgraded.
Need to get 473 kB of archives.
After this operation, 2,486 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all 0.11.2-6 [394 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify all 0.9.6-1.3 [24.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.13 [53.4 kB]
Fetched 473 kB in 1s (410 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 74837 files and directories currently installed.)
Preparing to unpack .../fail2ban_0.11.2-6_all.deb ...
Unpacking fail2ban (0.11.2-6) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../python3-pyinotify_0.9.6-1.3_all.deb ...
Unpacking python3-pyinotify (0.9.6-1.3) ...
Selecting previously unselected package whois.
Preparing to unpack .../whois_5.5.13_amd64.deb ...
Unpacking whois (5.5.13) ...
Setting up whois (5.5.13) ...
Setting up fail2ban (0.11.2-6) ...
Setting up python3-pyinotify (0.9.6-1.3) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...
```

Fail2ban will be set to ban a user after 5 failed attempts for 15 minutes.

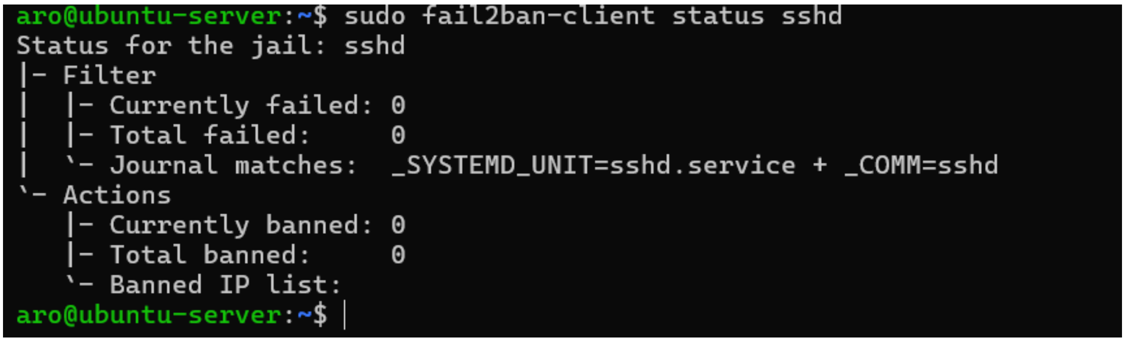


```
[sshd]

enabled = true
port    = ssh
backend = systemd
maxretry = 5
findtime = 180
bantime = 900
```

We can now confirm that Fail2ban is active and functioning as expected.

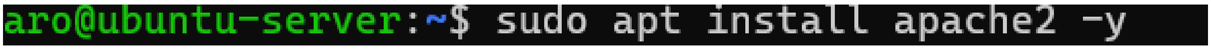


```
aro@ubuntu-server:~$ sudo systemctl restart fail2ban
aro@ubuntu-server:~$ sudo systemctl status fail2ban
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; disabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-11-10 08:54:32 UTC; 12s ago
       Docs: man:fail2ban(1)
   Main PID: 19161 (fail2ban-server)
      Tasks: 5 (limit: 4554)
     Memory: 12.2M
        CPU: 829ms
     CGroup: /system.slice/fail2ban.service
             └─19161 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Nov 10 08:54:32 ubuntu-server systemd[1]: Started Fail2Ban Service.
Nov 10 08:54:33 ubuntu-server fail2ban-server[19161]: Server ready
```

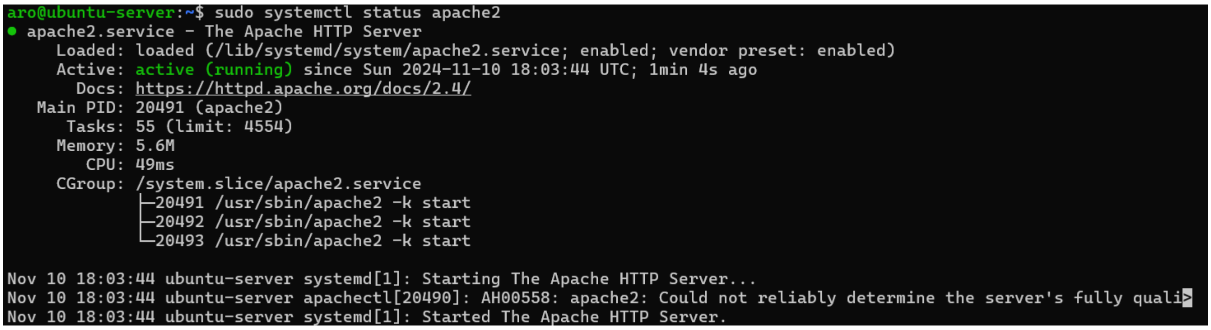


```
aro@ubuntu-server:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|  |- Currently failed: 0
|  |- Total failed:     0
|  `- Journal matches:  _SYSTEMD_UNIT=sshd.service + _COMM=sshd
`- Actions
   |- Currently banned: 0
   |- Total banned:     0
   `- Banned IP list:
aro@ubuntu-server:~$ 
```

We install the Apache2 web server and confirm that port 80 of the web server is working.

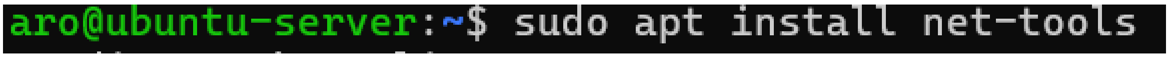


```
aro@ubuntu-server:~$ sudo apt install apache2 -y
```



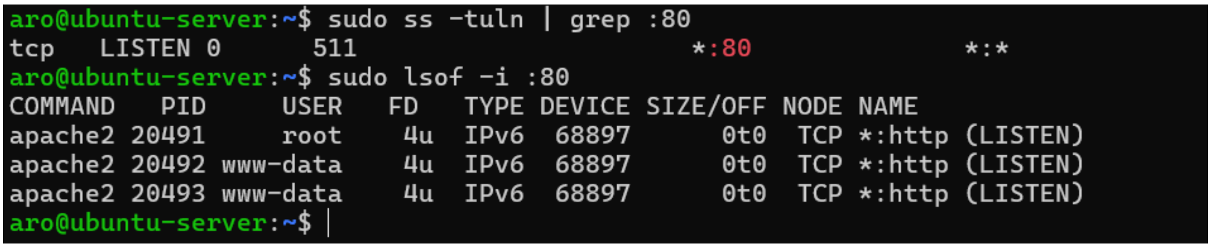
```
aro@ubuntu-server:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2024-11-10 18:03:44 UTC; 1min 4s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 20491 (apache2)
      Tasks: 55 (limit: 4554)
     Memory: 5.6M
        CPU: 49ms
     CGroup: /system.slice/apache2.service
             ├─20491 /usr/sbin/apache2 -k start
             ├─20492 /usr/sbin/apache2 -k start
             └─20493 /usr/sbin/apache2 -k start

Nov 10 18:03:44 ubuntu-server systemd[1]: Starting The Apache HTTP Server...
Nov 10 18:03:44 ubuntu-server apachectl[20490]: AH00558: apache2: Could not reliably determine the server's fully quali>
Nov 10 18:03:44 ubuntu-server systemd[1]: Started The Apache HTTP Server.
```

install the net-tools package, which includes the netstat command.



```
aro@ubuntu-server:~$ sudo apt install net-tools
```

All this shows that Apache2 is running successfully on our port 80 and is available for login.



```
aro@ubuntu-server:~$ sudo ss -tuln | grep :80
tcp   LISTEN 0      511                    *:80              *:*
aro@ubuntu-server:~$ sudo lsof -i :80
COMMAND   PID     USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
apache2 20491     root    4u  IPv6  68897      0t0  TCP *:http (LISTEN)
apache2 20492 www-data    4u  IPv6  68897      0t0  TCP *:http (LISTEN)
apache2 20493 www-data    4u  IPv6  68897      0t0  TCP *:http (LISTEN)
aro@ubuntu-server:~$ |
```