DID YOU MAKE THE CHANGES I ASKED FOR?
THAT DEPENDS.
www.dilbert.com scottadams@aol.com
DO YOU REMEMBER WHAT YOU ASKED ME TO CHANGE?
4-10-08 © 2008 Scott Adams, Inc./Dist. by UFS, Inc.
NO.
YUP, I MADE THE CHANGES.

#Be ready … :D.. #will start in next 2.. minutes

Remote System Management

Linux Administration

#Reference : tutorialspoint linux admin

Let's explore 2 methods for remote management in Linux system

- Console Management
- GUI Management

-

Remote Console Management

Remote Console Management means performing administration tasks from the command line via a service such as ssh. To use Linux effectively, as an Administrator, you will need to be proficient with the command line.

Linux at its heart was designed to be used from the console. Even today, some system administrators prefer the power of the command and save money on the hardware by running bare-bones Linux boxes with no physical terminal and no GUI installed.

Remote GUI Management

Remote GUI Management is usually accomplished in two ways: either a remote X-Session or a GUI application layer protocol like VNC. Each has its strengths and drawbacks.

However, for the most part, VNC is the best choice for Administration. It allows graphical control from other operating systems such as Windows or OS X that do not natively support the X Windows protocol.

Laying the Foundation for Security with SSH for Remote Console Access

ssh or Secure Shell is now the standard for remotely administering any Linux server. SSH unlike telnet uses TLS for authenticity and end-to-end encryption of communications. When properly configured an administrator can be pretty sure both their password and the server are trusted remotely.

Before configuring SSH, let's talk a little about the basic security and least common access. When SSH is running on its default port of 22; sooner rather than later, you are going to get brute force dictionary attacks against common usernames and passwords.

Following are a few rules of security to follow using SSH for remote administration on a production server −

- Never use a common username or password. Usernames on the system should not be system default, or associated with the company email address like: systemadmin@yourcompany.com
- Root access or administration access should not be allowed via SSH. Use a unique username and su to root or an administration account once authenticated through SSH.
- Password policy is a must: Complex SSH user passwords like: "This&IS&a&GUD&P@ssW0rd&24&me". Change passwords every few months to eliminate susceptibility to incremental brute force attacks.
- Disable abandoned or accounts that are unused for extended periods. If a hiring manager has a voicemail stating they will not be doing interviews for a month; that can lead to tech-savvy individuals with a lot time on their hands, for example.
- Watch your logs daily. As a System Administrator, dedicate at least 30-40 minutes every morning reviewing system and security logs. If asked, let everyone know you don't have the time to not be proactive. This practice will help isolate warning signs before a problem presents itself to end-users and company profits.

Install and Configure SSH for Remote Access

>_SSH

What is SSH and why it is required?

SSH stands for Secure Shell. It is a communication protocol, which helps us in communication with other devices over a network, just like HTTP does. So what’s the difference?

It is known for sending encrypted data over the network so that it can be prevented from unauthorized access. It runs on port number 22 by default. SSH first ensures the authenticity of the client and then build a pipeline between the SSH client and the server.

Data transmitted through this pipeline is encrypted by using the concept of Asymmetric Data Encryption

When to use SSH?

Following are the use cases for using SSH.

1. For transferring some data securely over the network.
2. Get access to a remote server.

OpenSSH is the premier connectivity tool for remote login with the SSH protocol. It encrypts all traffic to eliminate eavesdropping, connection hijacking, and other attacks. In addition, OpenSSH provides a large suite of secure tunneling capabilities, several authentication methods, and sophisticated configuration options.

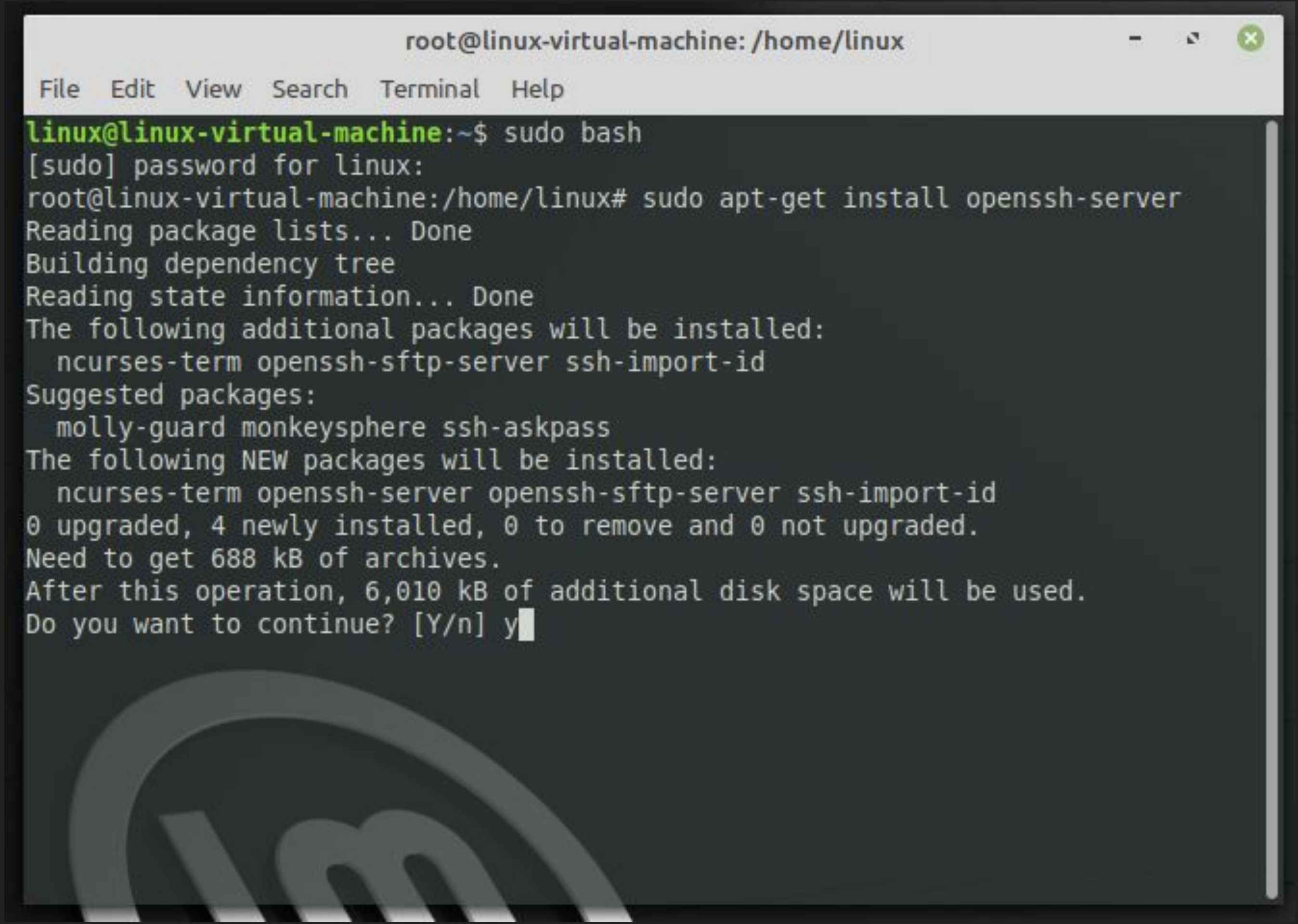
The OpenSSH suite consists of the following tools:

- Remote operations are done using ssh, scp, and sftp.
- Key management with ssh-add, ssh-keysign, ssh-keyscan, and ssh-keygen.
- The service side consists of sshd, sftp-server, and ssh-agent.
- OpenSSH is developed by a few developers of the OpenBSD Project and made available under a BSD-style license.

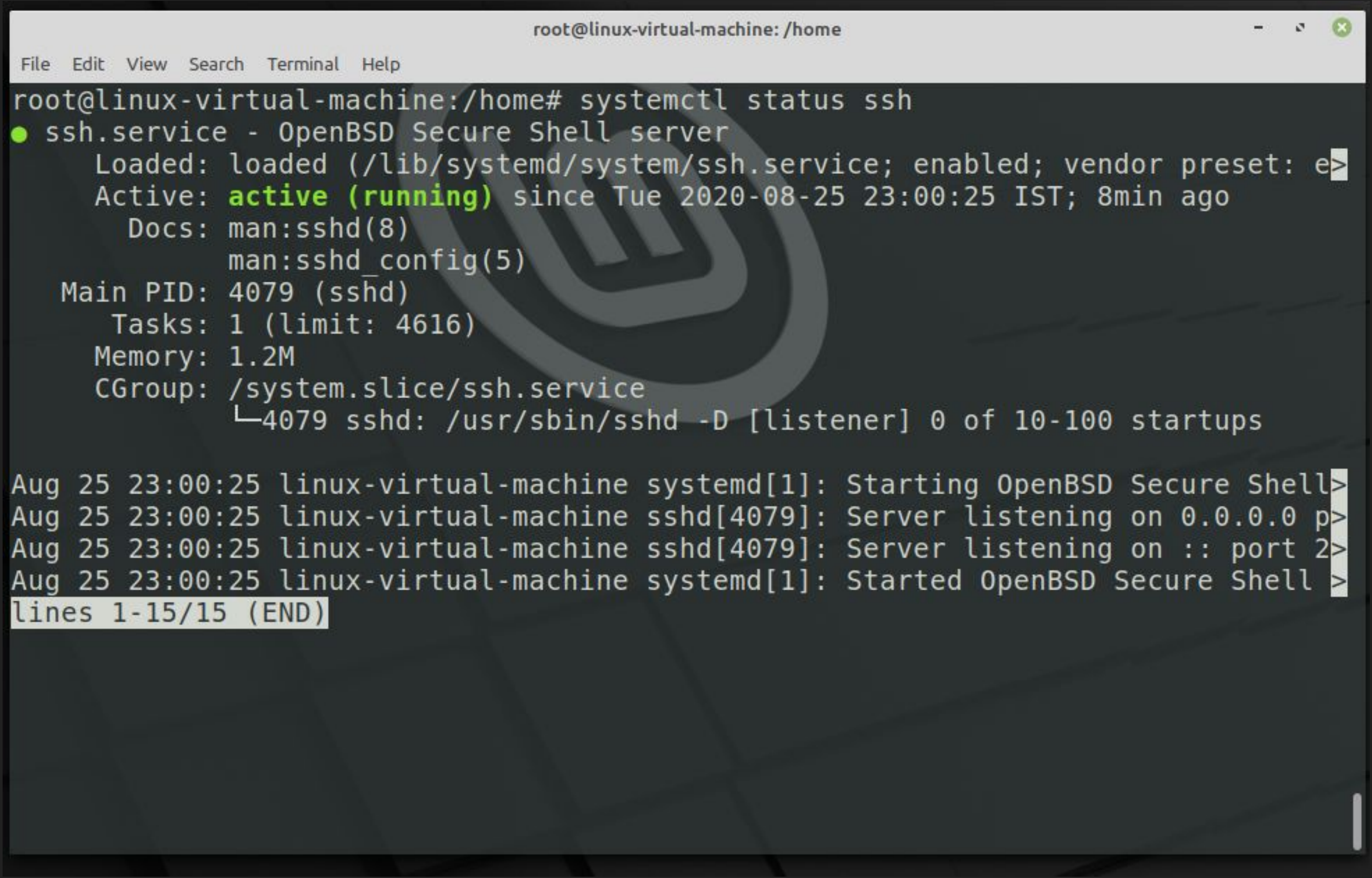
Following is the command to install SSHD: sudo apt-get install openssh-server

This will allows the machine to listen to ssh connections. Now we can get access of that remote machine by using the following command:

```
ssh <userid>@<IPaddress>
```

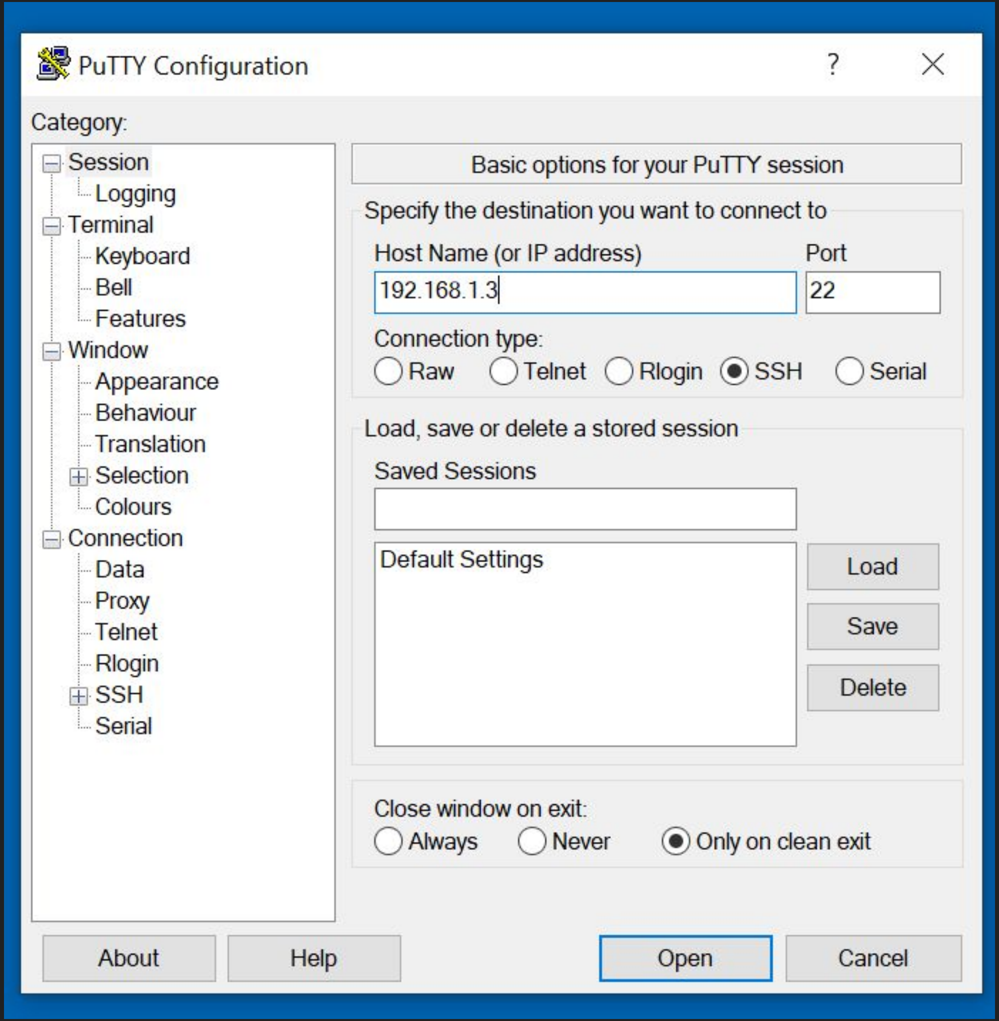

root@linux-virtual-machine: /home/linux
File Edit View Search Terminal Help

```
linux@linux-virtual-machine:~$ sudo bash
[sudo] password for linux:
root@linux-virtual-machine:/home/linux# sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
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 0 not upgraded.
Need to get 688 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```


root@linux-virtual-machine: /home
File Edit View Search Terminal Help

```
root@linux-virtual-machine:/home# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: e>
     Active: active (running) since Tue 2020-08-25 23:00:25 IST; 8min ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 4079 (sshd)
      Tasks: 1 (limit: 4616)
     Memory: 1.2M
     CGroup: /system.slice/ssh.service
             └─4079 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Aug 25 23:00:25 linux-virtual-machine systemd[1]: Starting OpenBSD Secure Shell>
Aug 25 23:00:25 linux-virtual-machine sshd[4079]: Server listening on 0.0.0.0 p>
Aug 25 23:00:25 linux-virtual-machine sshd[4079]: Server listening on :: port 2>
Aug 25 23:00:25 linux-virtual-machine systemd[1]: Started OpenBSD Secure Shell >
lines 1-15/15 (END)
```


PuTTY Configuration

Category:

- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - Telnet
  - Rlogin
  - SSH
  - Serial

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.3

Port: 22

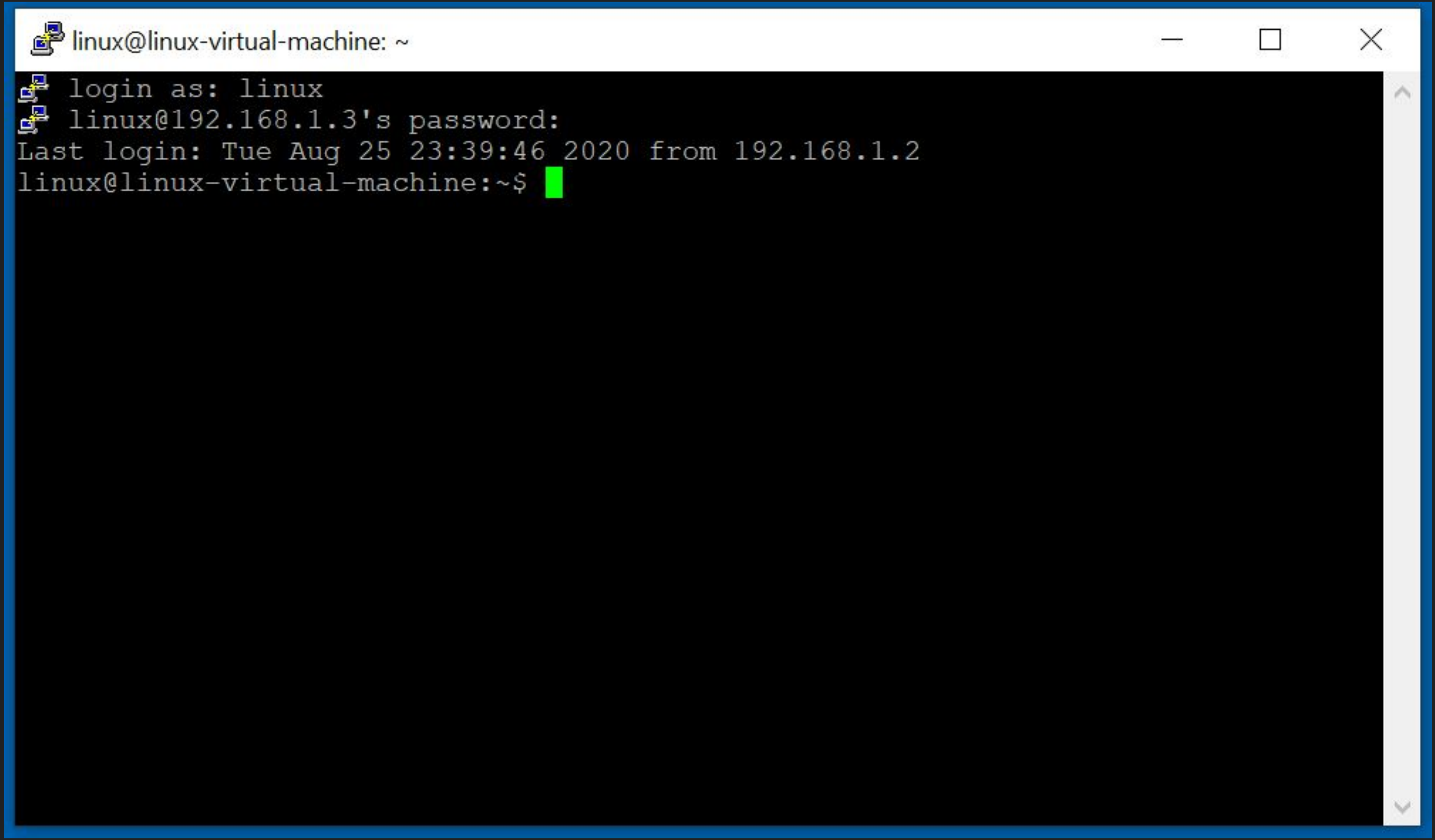
Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]



linux@linux-virtual-machine: ~

```
login as: linux
linux@192.168.1.3's password:
Last login: Tue Aug 25 23:39:46 2020 from 192.168.1.2
linux@linux-virtual-machine:~$
```

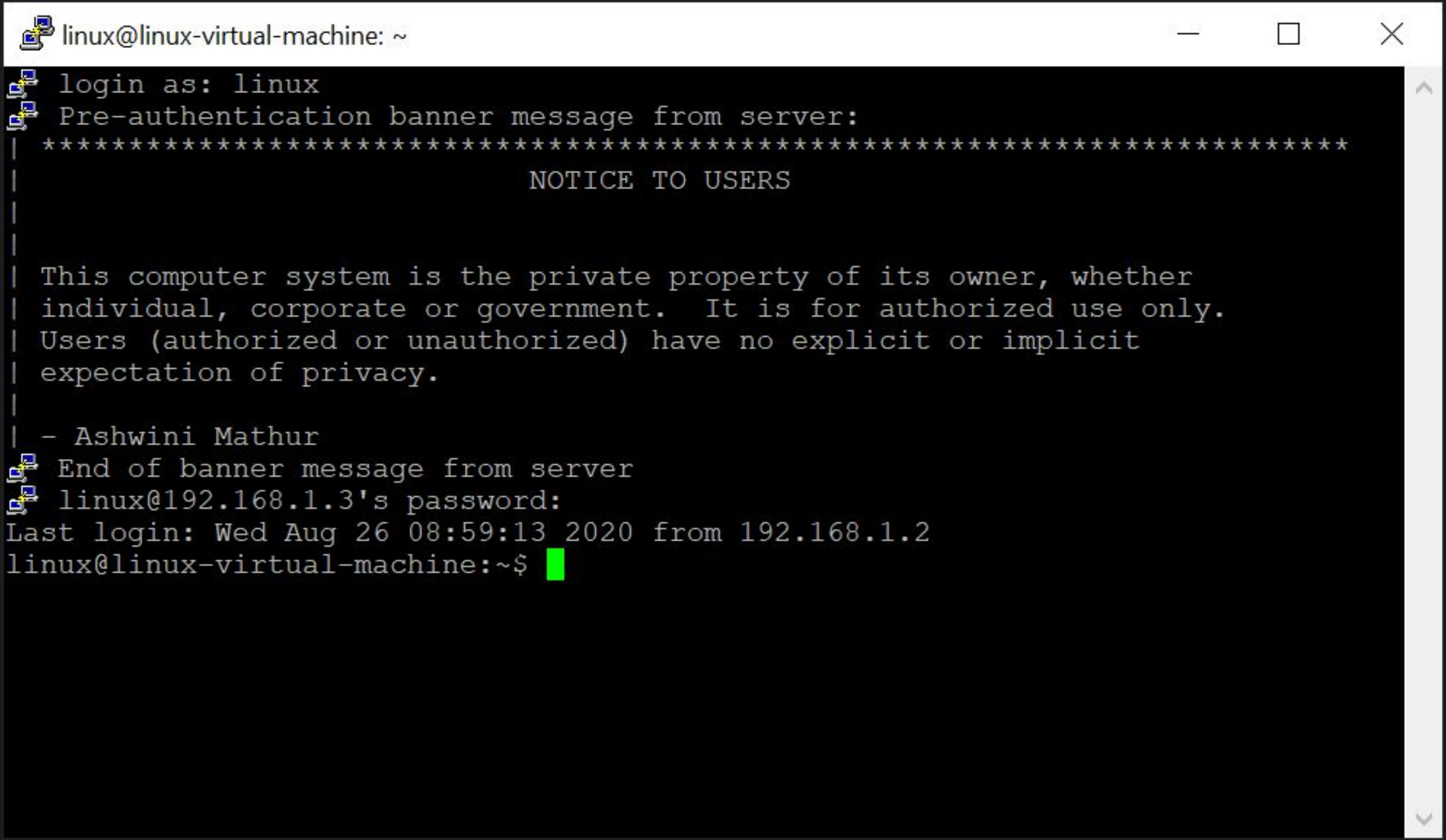
Banner

There are two way to display messages one is using issue.net file and second one is using MOTD file.

1. issue.net : Display a banner message before the password login prompt.
2. motd : Display a banner message after the user has logged in.

ssh with pre-configured banner message

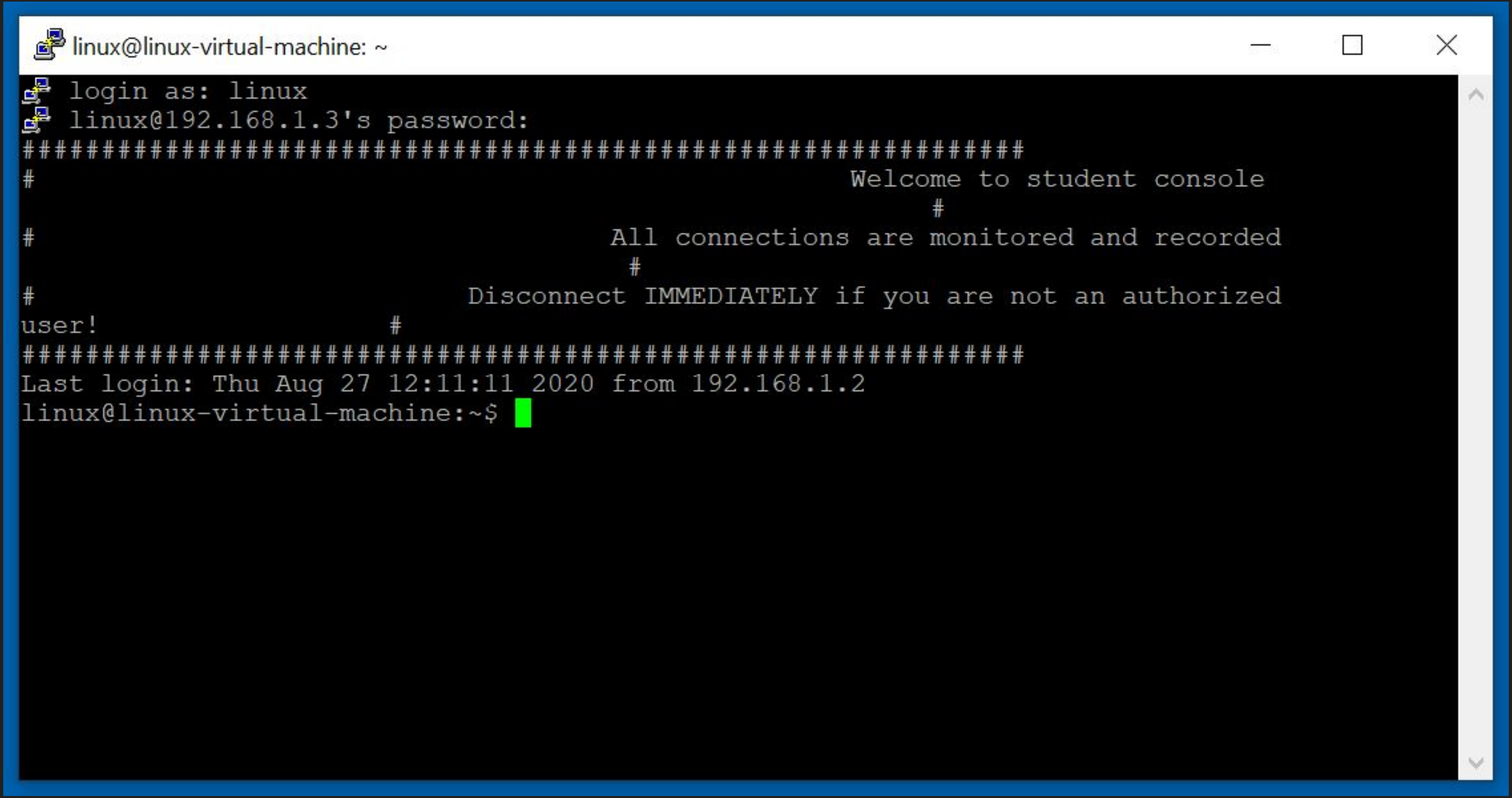
1. Edit file - sudo nano /etc/issue.net
2. Edit file :sudo nano /etc/ssh/sshd_config and uncomment line Banner/issue.net
3. restart the ssh service again : systemctl restart ssh



linux@linux-virtual-machine: ~

```
login as: linux
Pre-authentication banner message from server:
| ***************************************************************************
|                              NOTICE TO USERS
|
|
| This computer system is the private property of its owner, whether
| individual, corporate or government.  It is for authorized use only.
| Users (authorized or unauthorized) have no explicit or implicit
| expectation of privacy.
|
| - Ashwini Mathur
End of banner message from server
linux@192.168.1.3's password:
Last login: Wed Aug 26 08:59:13 2020 from 192.168.1.2
linux@linux-virtual-machine:~$
```

After login - for configuring the banner
Edit /etc/motd - then configure the banner text and save.
Restart ssh service



```
login as: linux
linux@192.168.1.3's password:
################################################################
#                                                     Welcome to student console
                                                          #
#                                     All connections are monitored and recorded
                                       #
#                            Disconnect IMMEDIATELY if you are not an authorized
user!                  #
################################################################
Last login: Thu Aug 27 12:11:11 2020 from 192.168.1.2
linux@linux-virtual-machine:~$
```

collections of these comics at: wizardzines.com

ssh

JULIA EVANS
@b0rk

♥ ssh keys ♥

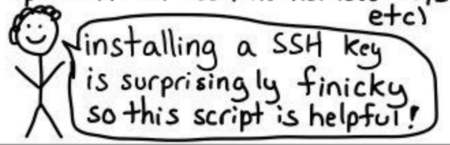
A ssh key is a secret key that lets you SSH to a machine without a password

ssh-copy-id

This script installs your SSH key on a host!

```
$ ssh-copy-id user@host
```

(puts it in .ssh/authorized_keys etc)


installing a SSH key is surprisingly finicky so this script is helpful!

★ port forwarding ★

```
ssh user@host.com -NfL
3333:localhost:8888
```

↑ local port
↑ remote port

Super helpful if you have a remote server you want to open in your browser!

just run 1 command

```
$ ssh user@host uname -a
```

runs this command & exits

~.

<Enter>~. closes the SSH connection. Useful if it's hanging!

.ssh/config

Lets you set, per host:

- username to use
- SSH key to use
- an alias!

so you can type `$ ssh ALIAS` instead of `ssh user@verylongdomain.com`

ssh-agent

remembers your SSH key passphrase so you don't have to keep typing it

mosh

Keeps a SSH connection open if your internet has a blip

ssh key based authentication

SSH key pair generation:

To generate a pair of RSA keys, the command is : ssh-keygen

The keys will be generated as follows:

- Private key: ~/.ssh/id_rsa
- Public key: ~/.ssh/id_rsa.pub

These generated keys will be encrypted using RSA cypher method. To use any other cypher technique, you need to use -t flag as follows: ssh-keygen -t dsa

Now for SSH authentication, we need to add our public key to remote machine’s authorized_hosts file. For this, we will use scp command which means secure copy.

```
scp ~/.ssh/id_rsa.pub userid@IPaddress:~/.ssh/authorized_keys
```

Configuration steps

Step One—Create the RSA Key Pair

The first step is to create the key pair on the client machine (there is a good chance that this will just be your computer):

```
ssh-keygen -t rsa
```

Step Two—Store the Keys and Passphrase

Once you have entered the Gen Key command, you will get a few more questions:

```
Enter file in which to save the key (/home/demo/.ssh/id_rsa):
```

You can press enter here, saving the file to the user home (in this case, my example user is called demo).

```
Enter passphrase (empty for no passphrase):
```

Step Three—Copy the Public Key

Once the key pair is generated, it’s time to place the public key on the server that we want to use.

You can copy the public key into the new machine’s authorized_keys file with the ssh-copy-id command. Make sure to replace the example username and IP address below.

```
ssh-copy-id demo@192.168.1.3
```