DevOps external course

Remote access. Client-server implementation of a secure terminal

Lecture 5.6

Module 5 **Linux Essentials**

Serge Prykhodchenko

Logical Volume Manager

In Linux, Logical Volume Manager (LVM) is a device mapper framework that provides logical volume management for the Linux kernel. Most modern Linux distributions are LVM-aware to the point of being able to have their root file systems on a logical volume.

Heinz Mauelshagen wrote the original LVM code in 1998, when he was working at Sistina Software, taking its primary design guidelines from the HP-UX's volume manager.

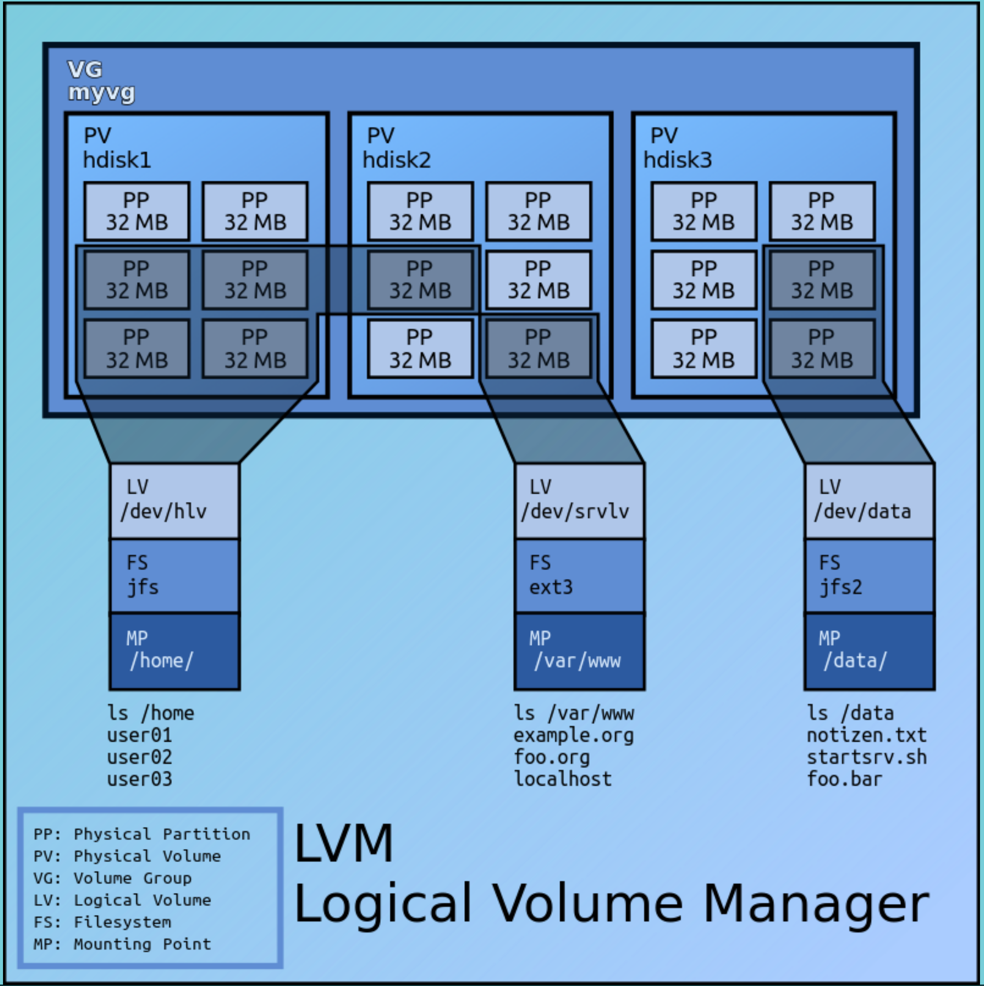
LVM is used for the following purposes:

- Creating single logical volumes of multiple physical volumes or entire hard disks (somewhat similar to RAID 0, but more similar to JBOD), allowing for dynamic volume resizing.
- Managing large hard disk farms by allowing disks to be added and replaced without downtime or service disruption, in combination with hot swapping.
- On small systems (like a desktop), instead of having to estimate at installation time how big a partition might need to be, LVM allows filesystems to be easily resized as needed.
- Performing consistent backups by taking snapshots of the logical volumes.
- Encrypting multiple physical partitions with one password.

LVM can be considered as a thin software layer on top of the hard disks and partitions, which creates an abstraction of continuity and ease-of-use for managing hard drive replacement, repartitioning and backup.

Logical Volume Manager

- Volume groups (VGs) can be resized online by absorbing new physical volumes (PVs) or ejecting existing ones.
- Logical volumes (LVs) can be resized online by concatenating extents onto them or truncating extents from them.
- LVs can be moved between PVs.
- Creation of read-only snapshots of logical volumes (LVM1), leveraging a copy on write (CoW) feature,or read/write snapshots (LVM2)
- VGs can be split or merged in situ as long as no LVs span the split. This can be useful when migrating whole LVs to or from offline storage.
- LVM objects can be tagged for administrative convenience.
- VGs and LVs can be made active as the underlying devices become available through use of the lvmetad daemon.



Group passwords

Group password – it’s a legacy field. (ru-wiki)

Here's a practical use for group passwords, that I implemented for myself on our work server, since the logs indicated my account was being brute-forced (or could have been a dictionary attack).
I used ssh-keygen and puttygen respectfully to generate key pairs for use from my workstation and home computer. The key I use from home requires a password. I added both of the public keys to the .ssh/authorized_keys, created a group marionette with a password and no members. As root I used visudo to add the following lines.

```
Cmnd_Alias      SUDOING = /bin/bash, /usr/bin/sudo -i
%marionette     ALL=NOPASSWD:SUDOING
```

I have disabled my account's password, you no one can log into it that way. I now login only with my keys and entering the password-protected group with newgrp marionette allows me to become root using sudo -i.
Without the NOPASSWD: option it will require your user account password. If it is disabled and this group does not have NOPASSWD, you will not be able to sudo -i. It will also require your user account password if your command list does not have /bin/bash or whatever shell your root is using by default.

https://unix.stackexchange.com/questions/93123/typical-use-case-for-a-group-password

Access Control Lists

File system access control is an essential part of UNIX and Linux systems, and therefore its improvement is the primary goal of the developers of these systems. Special attention has been paid to support for access control lists (ACLs) as a generalization of the traditional model of **user/group/“everyone else”** privileges set for multiple users and groups at once.
ACLs are part of the filesystem implementation, so the filesystem you are using must provide explicit support for them. Almost all UNIX and Linux file systems today support ACLs in one form or another.

SELinux: Linux Systems with Enhanced Security

The SELinux operating system (as a project) was developed by the US National Security Agency (NSA), but from the end of 2000 it was transferred to the open source developers. SELinux is included in the Linux kernel (since version 2.6) and therefore is currently available in most distributions (but often in a not fully functional state).

SELinux is an implementation of a mandatory access control (MAC) system in which all privileges are assigned by administrators.

In a MAC environment, users cannot delegate their rights and cannot set access control parameters on objects that they (users) own. And therefore, such an operating system is more suitable for nodes with special requirements

Pluggable Authentication Modules

Pluggable Authentication Modules (**PAM**) constitute an authentication technology, not an access control technology. In others in words, the PAM technology is designed to look for an answer not to the question: “Does the user have the right X perform operation Y? ”. Answer to the question:“ How do you know that this is really user X? ” PAM technology is an important link in the access control chain in most systems.

Kerberos cryptographic authentication network protocol

Like PAM, Kerberos is designed to solve problems of authentication, not access control. (It is named after the three-headed dog that protected the entrance to the realm of Hades - Cerberus, or Kerberos.) But if PAM can be called a structural shell for authentication, then Kerberos is a specific method of authentication. The Kerberos implementation uses a trusted (third party) server to perform network-wide authentication tasks. Instead of selfauthentication on your computer, you provide your credentials (tickets) to the Kerberos service, and it gives you cryptographic credentials that you can present to other services as proof of your identity.

QUOTING

Quotas

- https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Storage_Administration_Guide/ch-disk-quotas.html
- Quotas give us the ability to keep track of users' disk usage: both blocks (disk space) and inodes (number of files)
- `quota` rpm must be installed
- For both blocks and inodes, we quotas allow hard limits and soft limits:
  - Soft limit: user is allowed to exceed a soft limit, but they will be warned, and after a grace period, they cannot increase usage
  - Hard limit: user is never allowed to exceed the hard limit
- We enable quotas for a filesystem
- Quotas can be applied to users and/or groups
- System administrator can report on all users' disk usage status
- Each user can see their own disk usage status (quota information)

Turning quotas on (and off)

- Example: enabling quotas on /home (separate /home filesystem)
  - In `/etc/fstab`, add the `usrquota,grpquota` mount options for the file system mounted on the /home mount point
  - Initialize the quota database files for `/home` with the command

```
quotacheck -cug /home
```

    - `c`: don't read quota files, create new quota database files
    - `u`: do user quotas
    - `g`: do group quotas
  - Turn quotas on
  - `quotaon -vaug # turn quotas on`
    - `v`: display a message for each filesystem affected
    - `a`: turn quotas on for all automatically mounted file systems according to `/etc/fstab`
    - `u:` user quotas
    - `g:` group quotas
  - `repquota -a   # report on quotas`
  - Turn quotas off
  - `quotaoff -vaug # turn quotas off`
  - quotaoff -vaug; quotacheck –vaug; quotaon –vaug #single user mode

Setting Quotas

- To set a quota for a user, as root

```
edquota username
```

  - where
    - you'll see (example) DO NOT edit blocks or inodes, just soft and hard limits!

Disk quotas for user tgk (uid 107):

Filesystem	blocks	soft	hard	inodes	soft	hard
/dev/sda8	108	1000	2000	1	0	0

or this command can be used in scripts

```
setquota -u username soft hard isoft ihard  fs
```

  - where
    - username is the name of the user
    - soft is the block soft limit
    - hard is the block hard limit
    - isoft is the inode soft limit
    - ihard is the inode hard limit
    - fs is the file system mount point (e.g. /home)

Quota Grace Period

- To set the grace period for all users

```
edquota -t     # edit grace period
```

  - where you'll see something like this (note units)

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem                                Block grace period     Inode grace period
/dev/mapper/VolGroup00-LogVol00                 8days                  8days
```

- To set the grace period for an individual user

```
edquota -T tgk
```

- where you'll see something like this (note units)

Times to enforce softlimit for user tgk (uid 498):
Time units may be: days, hours, minutes, or seconds

Filesystem	block grace	inode grace
/dev/mapper/VolGroup00-LogVol00	unset	unset

quota and repquota commands

- individual users can check their individual quota status with `quota` command:
  - shows
    - block usage and limits
    - inode usage and limits
    - remainder on grace period if over soft limit
- System administrator can print report of all users quota status (see also `warnquota`):
  - `repquota -a`
  - shows for each user what they've used, soft limits, hard limits, and remainder of grace periods if that user has entered one of their grace periods

Growing a filesystem

- That LVM tutorial link again:
  - http://www.howtoforge.com/linux_lvm
- When a file system resides on a LVM Logical Volume, we can
  - add a hard disk
  - create a partition on that hard disk
  - # or, maybe we already had an unused partition, such as a reclaimed Windows partition
  - set up that partition as a physical volume
  - add that physical volume to the Volume Group where that Logical Volume resides
  - grow the Logical Volume on the Volume Group
  - grow the file system on that Logical Volume

Growing a file system (cont'd)

- set up our "new" or "spare" partition as a physical volume for LVM (suppose it's /dev/sdb1):
  - pvcreate /dev/sdb1
- Add this new physical volume to a volume group (in this case VolGroup00):
  - vgextend VolGroup00 /dev/sdb1
- See how many free extents (Free PE) are available in this volume group (VolGroup00)
  - vgdisplay

Growing a file system (cont'd)

- Suppose the previous "vgdisplay" command showed that VolGroup00 had 319 free extents ("Free PE") and we use them all:
  - lvextend –l+319 /dev/VolGroup00/LogVol00
- Now LogVol00, which contains our root file system, is bigger, but the files ystem is still the same size.
- Grow the filesystem (ext4) to fill the added space (even if the file system is mounted):
  - resize2fs /dev/VolGroup00/LogVol00
- Use df command so see we have bigger file system now!

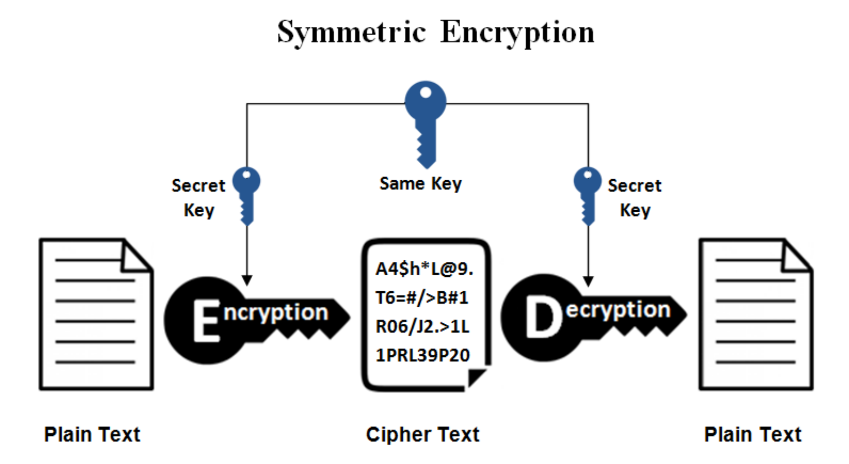
Challenges

1. Secure client-server connection
2. Automation

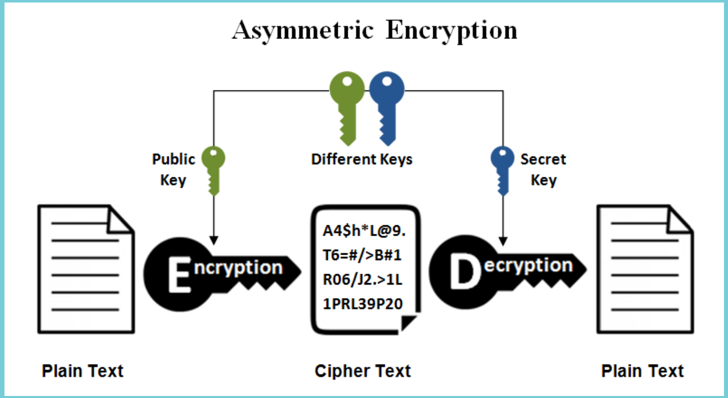
Secure client-server connection

Asymmetric cryptography vs symmetric

- This is the simplest kind of encryption that involves only one secret key to cipher and decipher information. Symmetrical encryption is an old and best-known technique. It uses a secret key that can either be a number, a word or a string of random letters. It is a blended with the plain text of a message to change the content in a particular way. The sender and the recipient should know the secret key that is used to encrypt and decrypt all the messages.
- Blowfish, AES, RC4, DES, RC5, and RC6 are examples of symmetric encryption. The most widely used symmetric algorithm is AES-128, AES-192, and AES-256.
- The main disadvantage of the symmetric key encryption is that all parties involved have to exchange the key used to encrypt the data before they can decrypt it.



Secure client-server connection



- Asymmetrical encryption is also known as public key cryptography, which is a relatively new method, compared to symmetric encryption. Asymmetric encryption uses two keys to encrypt a plain text. Secret keys are exchanged over the Internet or a large network. It ensures that malicious persons do not misuse the keys. It is important to note that anyone with a secret key can decrypt the message and this is why asymmetrical encryption uses two related keys to boosting security. A public key is made freely available to anyone who might want to send you a message. The second private key is kept a secret so that you can only know.
- A message that is encrypted using a public key can only be decrypted using a private key, while also, a message encrypted using a private key can be decrypted using a public key. Security of the public key is not required because it is publicly available and can be passed over the internet. Asymmetric key has a far better power in ensuring the security of information transmitted during communication.
- Asymmetric encryption is mostly used in day-to-day communication channels, especially over the Internet. Popular asymmetric key encryption algorithm includes ElGamal, RSA, DSA, Elliptic curve techniques, PKCS

Secure client-server connection

Asymmetric cryptography vs symmetric

- Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetrical encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating.
- Symmetric encryption is an old technique while asymmetric encryption is relatively new.
- Asymmetric encryption was introduced to complement the inherent problem of the need to share the key in symmetrical encryption model, eliminating the need to share the key by using a pair of publicprivate keys.
- Asymmetric encryption takes relatively more time than the symmetric encryption.

Secure client-server connection

telnet, rlogin, RDS vs ssh

Key Differences Between Telnet and SSH

- Telnet and SSH both serve the same purpose and provides the connectivity to the remote server but Telnet is conventional protocol, although it is still in use in the various application. SSH is the replacement for Telnet and has some enhanced features too.
- Telnet doesn’t provide any security mechanism whereas SSH is more secure and provides security measures.
- In Telnet transmits data in plain text that is the reason it is vulnerable to security attacks. On the other hand, SSH uses encryption for transmitted data and security breach does not likely occur. SSH can withstand eavesdropping, man in the middle and insertion/ replay attacks.
- Telnet doesn’t provide authentication facility while SSH provides user authentication.
- Telnet works with a private network. In contrast, SSH works with a public network.
- Telnet communicates via port number 23 over TCP/IP. As against, SSH uses port number 22 for communication.

Secure client-server connection

telnet, rlogin, RDS vs ssh

Key differences between Rlogin and SSH

- SSH traffic is encrypted while Rlogin traffic is not
- SSH authenticates the user while Rlogin does not
- SSH can be used for automation while Rlogin cannot
- Rlogin is no longer being used in favor of SSH

Secure client-server connection

telnet, rlogin, RDS vs ssh

Key differences between RDS and SSH

+ The Remote Desktop Protocol is solely used for accessing Windows virtual machines (VMs) and physical Windows servers (as opposed to Linux® servers). From a user perspective, RDP provides a Windows Graphical User Interface (GUI) experience, making servers more accessible to a wider range of employees — with or without a technical background.

- Because RDP ports often need to be connected to the internet for remote access, for security purposes, admins should protect their RDP instances with a virtual private network (VPN) and/or a form of multi-factor authentication (MFA). RDP ports can be vulnerable to attacks when exposed to the internet.

Automation

- Automated your SSH login with Public Key Authentication
- Using native Linux-based OS shell scripting language (sh, bash, etc)
- Automated scripts over SSH
- https://www.openssh.com/

Description

- Completely open source project with free licensing
- Strong cryptography (AES, ChaCha20, RSA, ECDSA, Ed25519...)

Encryption is started before authentication, and no passwords or other information is transmitted in the clear. Encryption is also used to protect against spoofed packets. A number of different ciphers and key types are available, and legacy options are usually phased out in a reasonable amount of time.

- X11 forwarding (which also encrypts X Window System traffic)

X11 forwarding allows the encryption of remote X windows traffic, so that nobody can snoop on your remote xterms or insert malicious commands. The program automatically sets DISPLAY on the server machine, and forwards any X11 connections over the secure channel. Fake Xauthority information is automatically generated and forwarded to the remote machine; the local client automatically examines incoming X11 connections and replaces the fake authorization data with the real data (never telling the remote machine the real information).

- Port forwarding (encrypted channels for legacy protocols)

Port forwarding allows forwarding of TCP/IP connections to a remote machine over an encrypted channel. Insecure internet applications like POP can be secured with this.

- Strong authentication (public keys, one-time passwords)

Strong authentication protects against several security problems: IP spoofing, fakes routes and DNS spoofing. Some authentication methods include public key authentication, one-time passwords with s/key and authentication using Kerberos (only in-portable)

Description

Agent forwarding

An authentication agent, running in the user's laptop or local workstation, can be used to hold the user's authentication keys. OpenSSH automatically forwards the connection to the authentication agent over any connections, and there is no need to store the authentication keys on any machine in the network (except the user's own local machine). The authentication protocols never reveal the keys; they can only be used to verify that the user's agent has a certain key. Eventually the agent could rely on a smart card to perform all authentication computations.

Interoperability

Interoperability between implementations is a goal, but not a promise. As OpenSSH development progresses, older protocols, ciphers, key types and other options that have known weaknesses are routinely disabled.

SFTP client and server support.

Complete SFTP support is included.

Optional data compression

Data compression before encryption improves the performance for slow network links

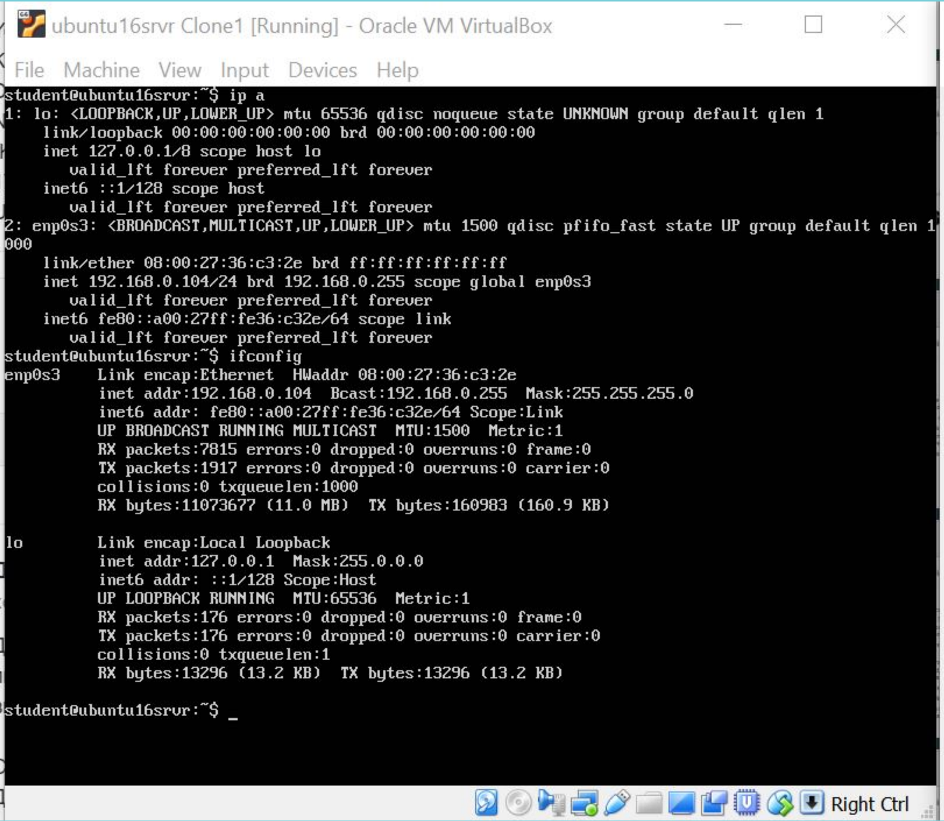
Realization. Prerequisites

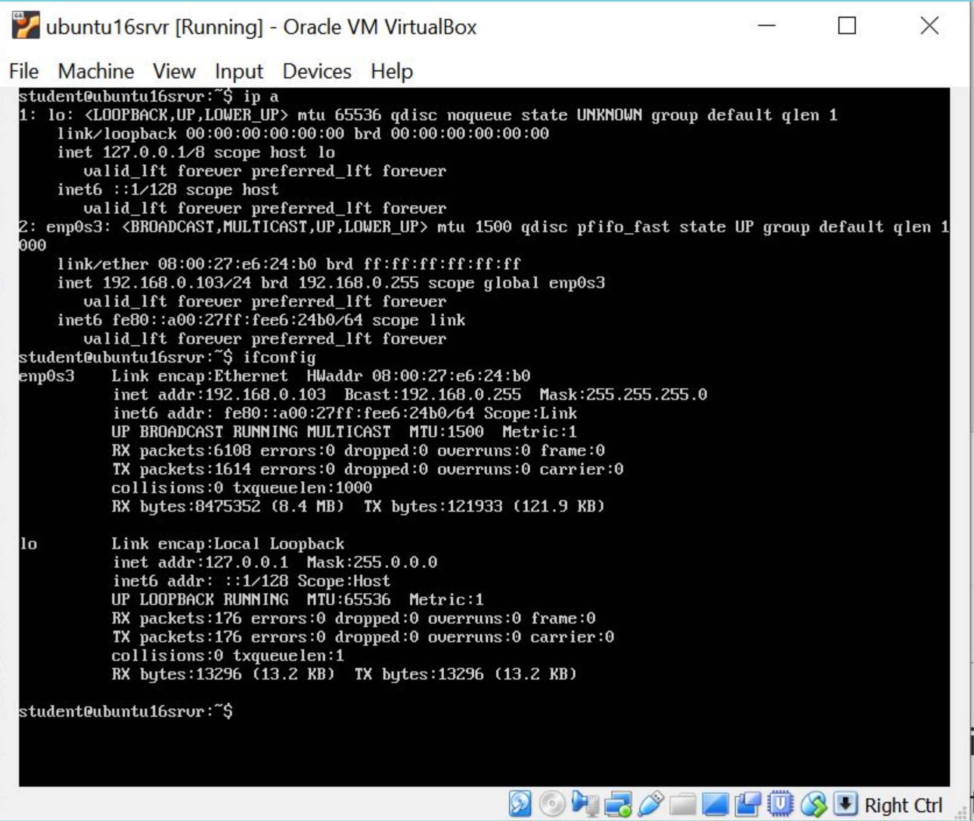
- VirtualBox ver. 5+
- Installed Ubuntu 16.04 with SSH server (server edition) with
“Network”-> “Adapter” set to “Bridged Adapter”
- Clone(-s) of Installed Ubuntu 16.04 with SSH server (server edition) with
“Network”-> “Adapter” set to “Bridged Adapter” (full clone with reinitialized MAC-addresses)

SSH client :
https://mobaxterm.mobatek.net/download.html (for Windows)
or
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html (Windows, Linux, but simple)
or

- https://www.termius.com/ (Windows, Linux, Mac)

Realization (1)





Realization (2)

ubuntu16srvr Clone1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
student@ubuntu16srvr:~$ ssh student@192.168.0.103
The authenticity of host '192.168.0.103 (192.168.0.103)' can't be established.
ECDSA key fingerprint is SHA256:/z3Q8P7qBEewbHT2topjj/Vgfz9w4mT4idf+gx+jK7E.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.103' (ECDSA) to the list of known hosts.
student@192.168.0.103's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-131-generic x86_64)

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

196 packages can be updated.
143 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Tue Aug 18 16:48:56 2020
student@ubuntu16srvr:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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 pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:e6:24:b0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.103/24 brd 192.168.0.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee6:24b0/64 scope link
       valid_lft forever preferred_lft forever
student@ubuntu16srvr:~$
```

Right Ctrl

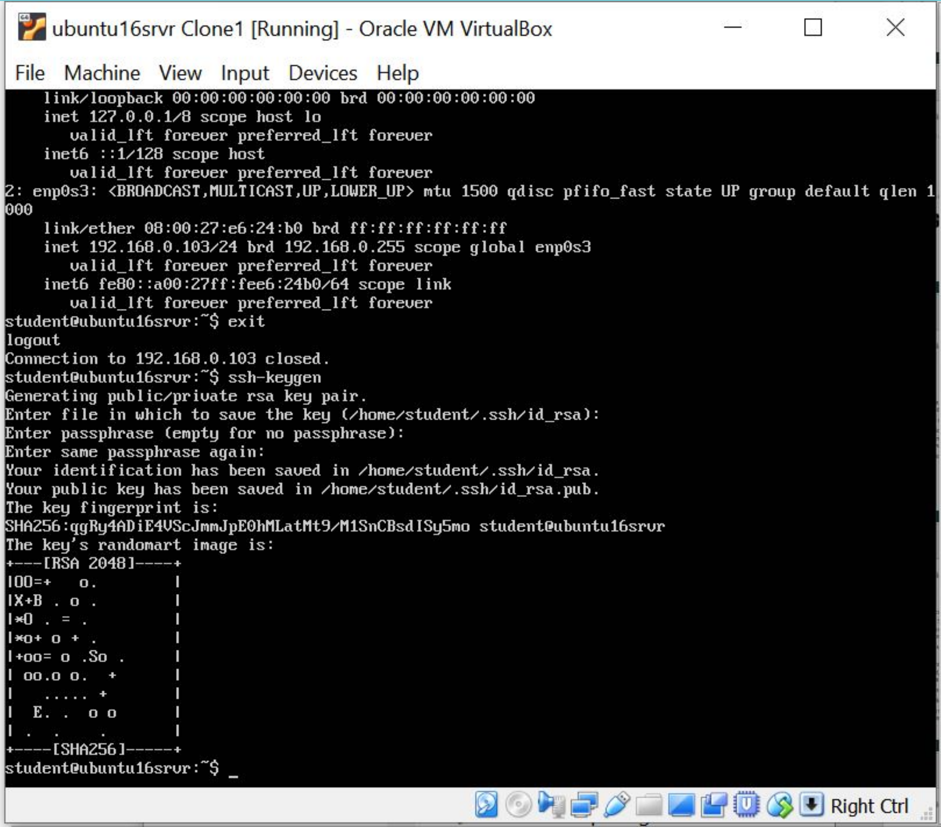
ubuntu16srvr [Running] - Oracle VM VirtualBox

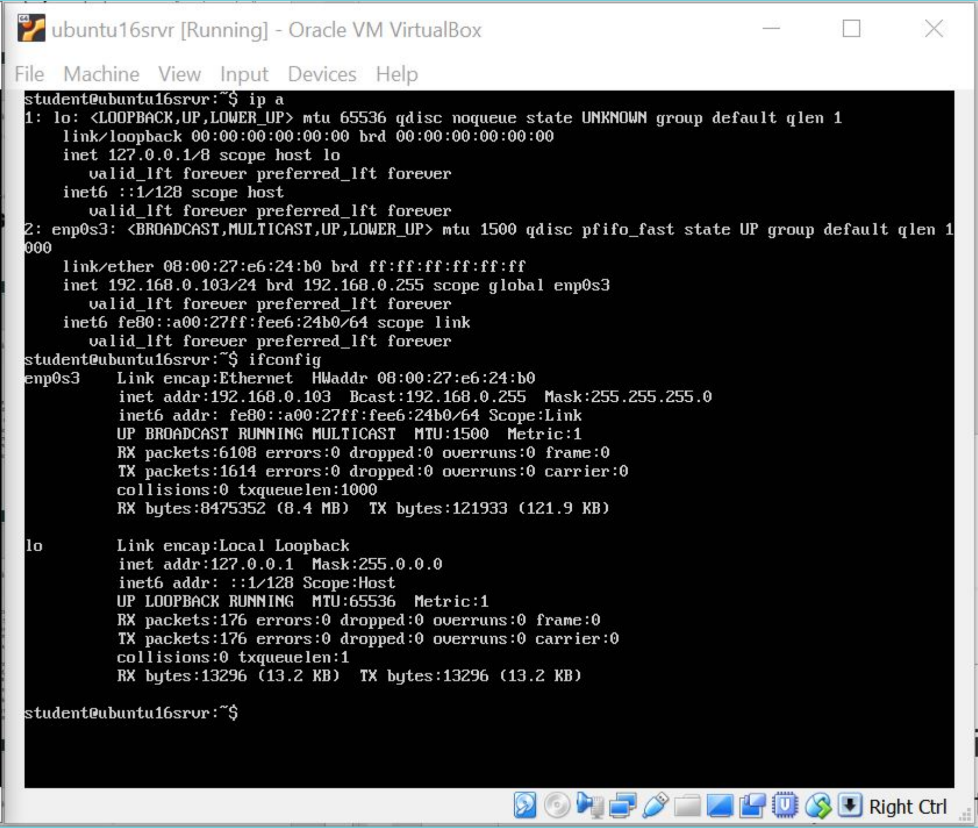
File Machine View Input Devices Help

```
student@ubuntu16srvr:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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 pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:e6:24:b0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.103/24 brd 192.168.0.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee6:24b0/64 scope link
       valid_lft forever preferred_lft forever
student@ubuntu16srvr:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:e6:24:b0
          inet addr:192.168.0.103  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee6:24b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8475352 (8.4 MB)  TX bytes:121933 (121.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:13296 (13.2 KB)  TX bytes:13296 (13.2 KB)

student@ubuntu16srvr:~$
```

Right Ctrl

Realization (3)





Realization (4)

ubuntu16srvr Clone1 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
student@ubuntu16srvr:~$ ssh-copy-id student@192.168.0.103
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/student/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
student@192.168.0.103's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'student@192.168.0.103'"
and check to make sure that only the key(s) you wanted were added.

student@ubuntu16srvr:~$ ssh student@192.168.0.103
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-131-generic x86_64)

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

196 packages can be updated.
143 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Tue Aug 18 17:00:17 2020 from 192.168.0.104
student@ubuntu16srvr:~$ _
```

Right Ctrl

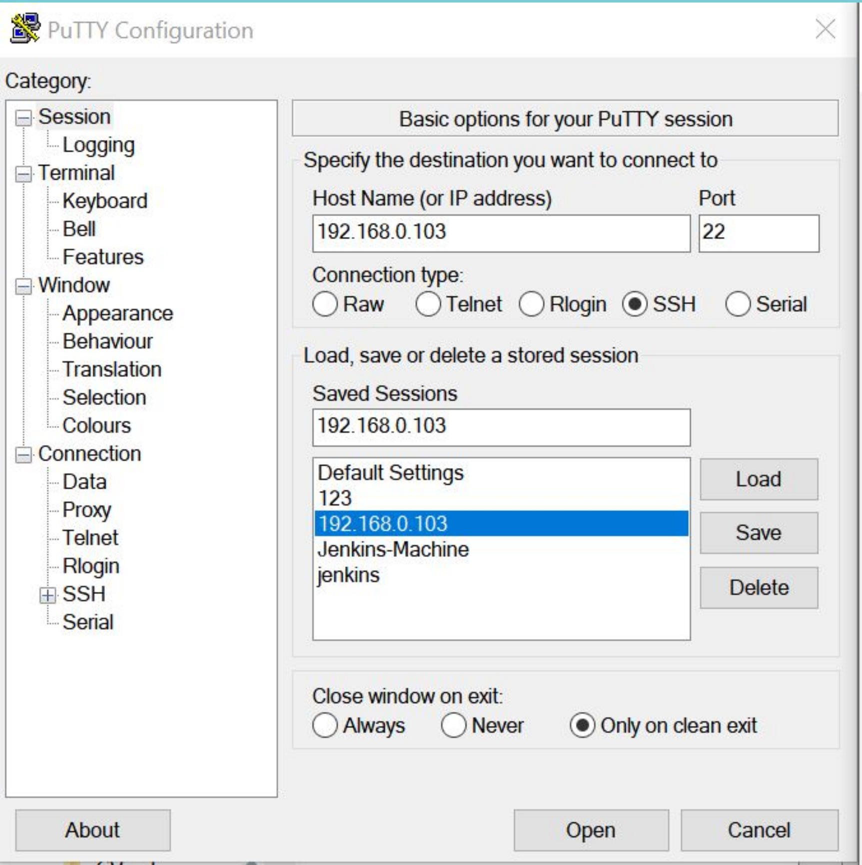
ubuntu16srvr [Running] - Oracle VM VirtualBox

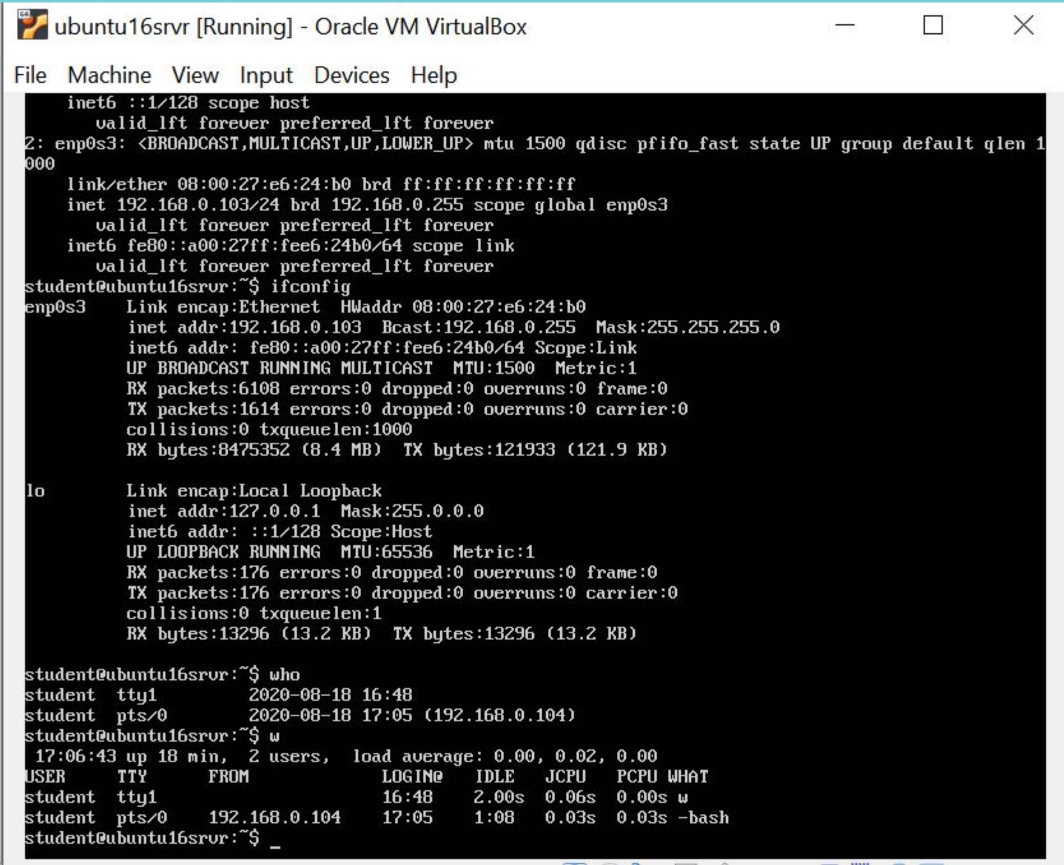
File Machine View Input Devices Help

```
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:e6:24:b0 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.103/24 brd 192.168.0.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee6:24b0/64 scope link
       valid_lft forever preferred_lft forever
student@ubuntu16srvr:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:e6:24:b0
          inet addr:192.168.0.103  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee6:24b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8475352 (8.4 MB)  TX bytes:121933 (121.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:13296 (13.2 KB)  TX bytes:13296 (13.2 KB)

student@ubuntu16srvr:~$ who
student  tty1         2020-08-18 16:48
student  pts/0        2020-08-18 17:05 (192.168.0.104)
student@ubuntu16srvr:~$ w
 17:06:43 up 18 min,  2 users,  load average: 0.00, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      16:48    2.00s  0.06s  0.00s w
student  pts/0    192.168.0.104    17:05    1:08   0.03s  0.03s -bash
student@ubuntu16srvr:~$ _
```

Right Ctrl

Realization (5)





Realization (6)

student@ubuntu16srvr: ~

```
login as: student
student@192.168.0.103's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-131-generic x86_64)

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

196 packages can be updated.
143 updates are security updates.

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade it.


Last login: Tue Aug 18 17:05:35 2020 from 192.168.0.104
student@ubuntu16srvr:~$
```

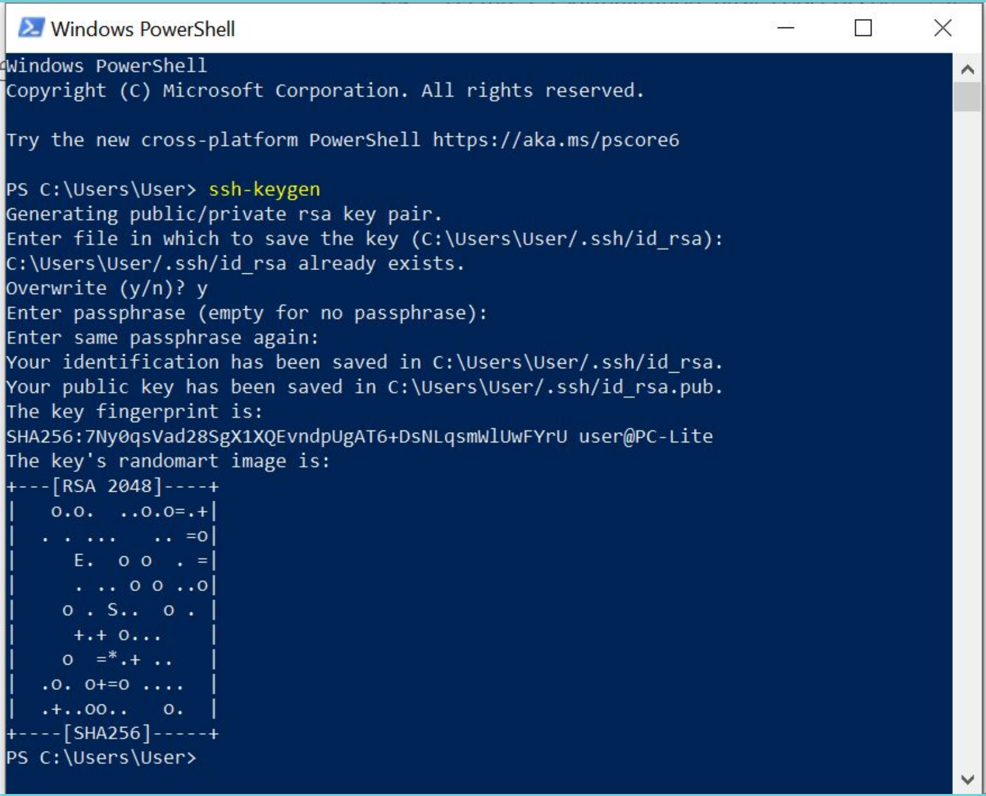
ubuntu16srvr [Running] - Oracle VM VirtualBox

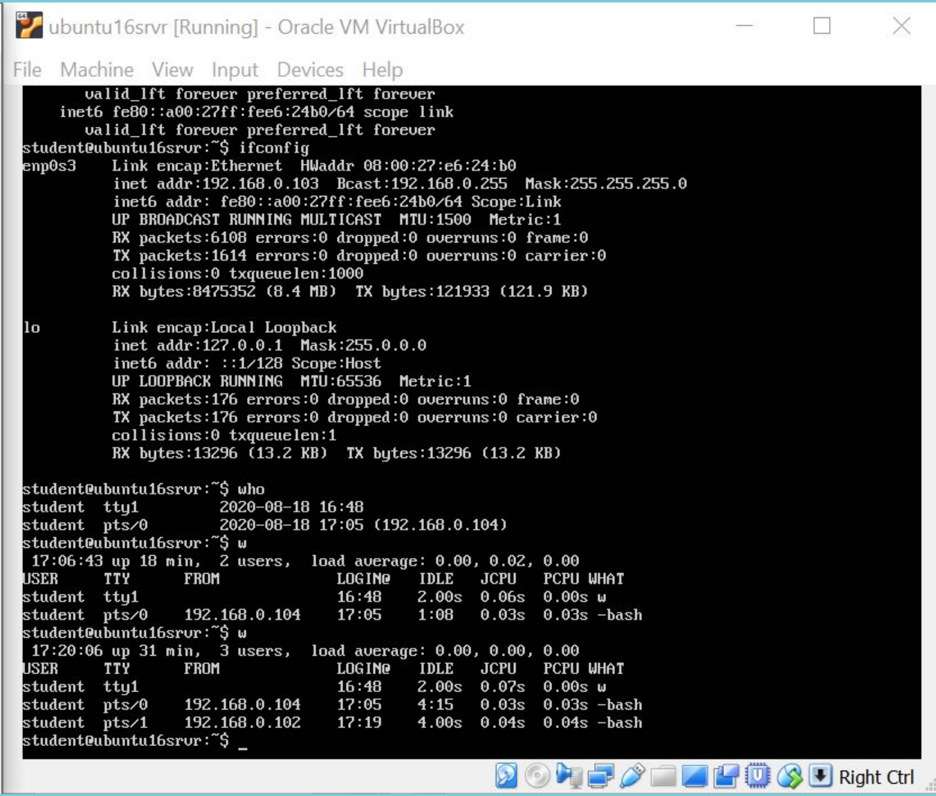
File Machine View Input Devices Help

```
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee6:24b0/64 scope link
       valid_lft forever preferred_lft forever
student@ubuntu16srvr:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:e6:24:b0
          inet addr:192.168.0.103  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fee6:24b0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8475352 (8.4 MB)  TX bytes:121933 (121.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:176 errors:0 dropped:0 overruns:0 frame:0
          TX packets:176 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:13296 (13.2 KB)  TX bytes:13296 (13.2 KB)

student@ubuntu16srvr:~$ who
student  tty1         2020-08-18 16:48
student  pts/0        2020-08-18 17:05 (192.168.0.104)
student@ubuntu16srvr:~$ w
 17:06:43 up 18 min,  2 users,  load average: 0.00, 0.02, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      16:48    2.00s  0.06s  0.00s w
student  pts/0    192.168.0.104    17:05    1:08   0.03s  0.03s -bash
student@ubuntu16srvr:~$ w
 17:20:06 up 31 min,  3 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      16:48    2.00s  0.07s  0.00s w
student  pts/0    192.168.0.104    17:05    4:15   0.03s  0.03s -bash
student  pts/1    192.168.0.102    17:19    4.00s  0.04s  0.04s -bash
student@ubuntu16srvr:~$ _
```

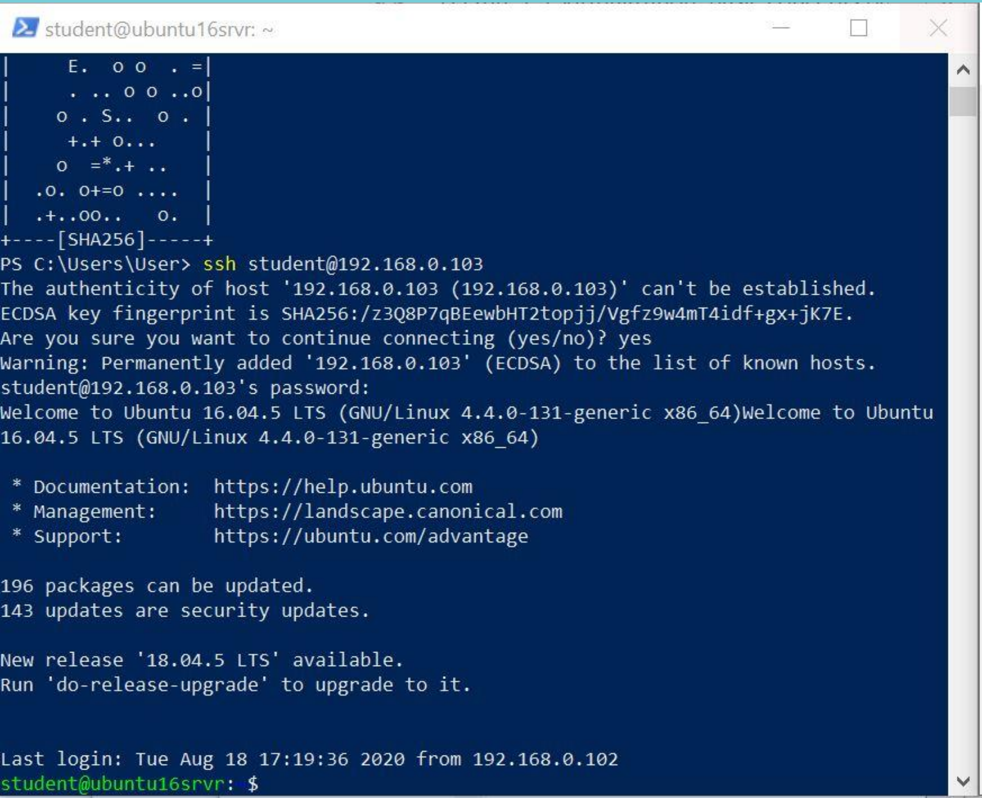
Right Ctrl

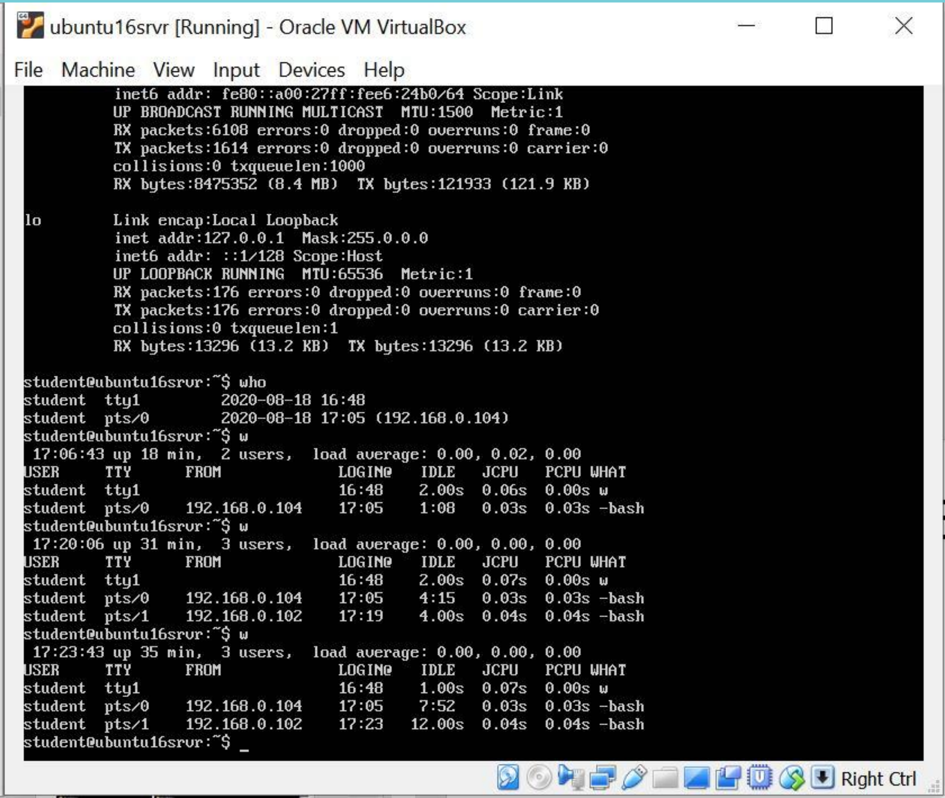
Realization (7)



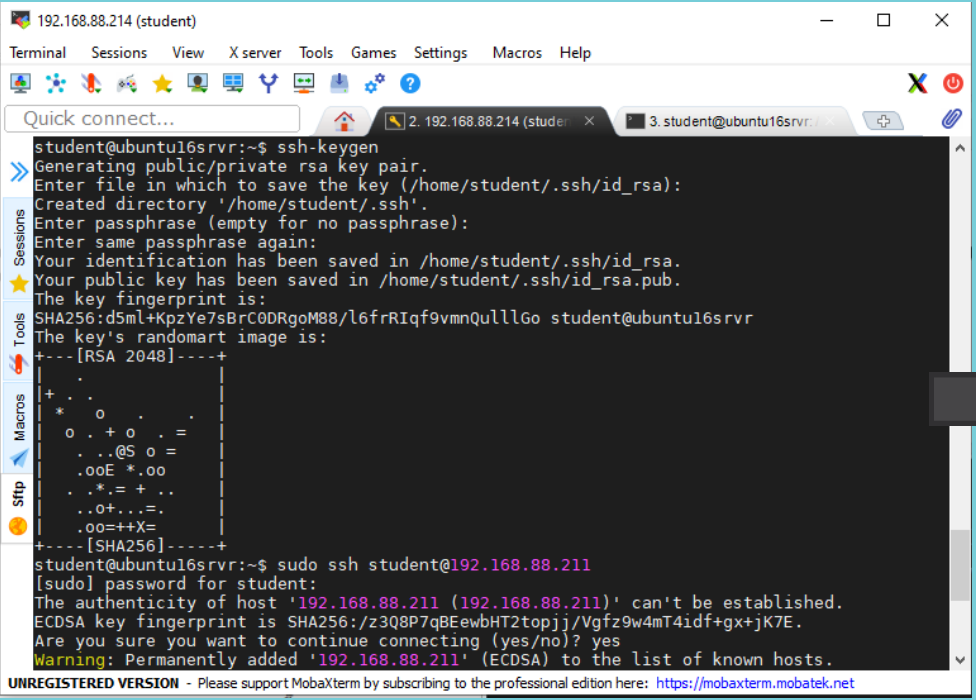


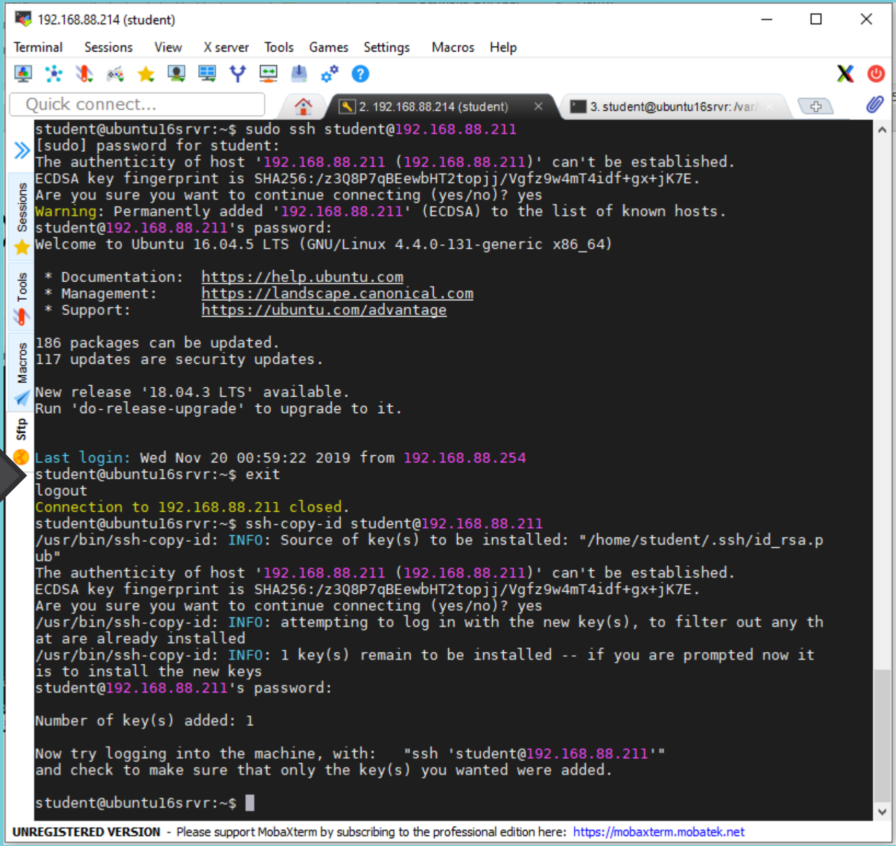
Realization (8)





Establish SSH connection without login/password (using MobaXTerm)

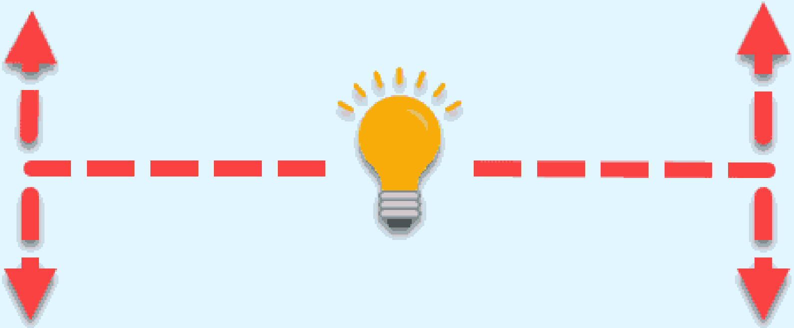




Additional settings. Change the port

The port you choose must not be in conflict with another application. If you select a port that is reserved for another service, you can run into issues. To make a decision, refer to the list of TCP and UDP port numbers assigned by the Internet Assigned Numbers Authority (IANA).

Only root users can listen on ports below 1024.



Verify that the port you decide to use is not blocked.

Avoid the most common variations of port 22, such as 222, 2222, and 22222.

Additional settings. Change the port

nano /etc/ssh/sshd_config

```
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes
```

service sshd restart

Additional settings. Disable Server SSH Root Login

- Use SSH to log into the server as root.
- Use a text editor to open the main configuration file. This time, we will use the vi editor.

vi /etc/ssh/sshd_config

- Find the line that says “PermitRootLogin_yes“ and change to PermitRootLogin_no. You may need to scroll down a few lines to find it.
- It is important to add the user account you will use to log in. Just add another line with the username in question: AllowUsers your_username_here
- Save the changes you made and then exit the text editor.
- Restart the SSH service but do not close the root session yet. For Ubuntu/Debian use sudo service ssh restart and for Fedora/CentOS use the service ssh restart command.
- Open a new terminal window and verify that you can now log in as the user you added. Once you confirm it works, exit the active root session.

Additional settings. Disable Password-Based Logins on Your Server

- Use SSH keys to log into the server as root or with sudo privileges.
- Use a text editor to open the sshd_config file. We will use vi:

vi /etc/ssh/sshd_config

- Look for the line that says PasswordAuthentication and change to

PasswordAuthentication_no.

Make sure to uncomment the line if the # is present.

- Save the changes you’ve made and then exit the text editor.
- Restart the SSH service to apply the changes. For Ubuntu/Debian use sudo service ssh restart and for Fedora/CentOS use the service ssh restart command

Additional settings. Restrict SSH Access Using iptables

- Iptables is a Linux utility used for configuring firewall rules and monitoring/filtering incoming and outgoing traffic to your server. It is included by default with most Linux distributions.
- With iptables, you can define rules that limit or permit traffic for different kinds of services by IP address, port or network protocol and thus substantially improve the security of your server. In our case, we will set firewall rules to restrict the incoming SSH traffic for everyone but one IP address or subnet.
- This way, blocking port 22 will not only stop unauthorized access to your servers but can also stop or prevent DDoS attacks.
- While taking this step, you should make sure you do not lock yourself out by completely blocking SSH traffic. You will need to use only a few commands to allow a specific IP address or subnet for incoming SSH connections.

QUESTIONS & ANSWERS

THANK YOU!