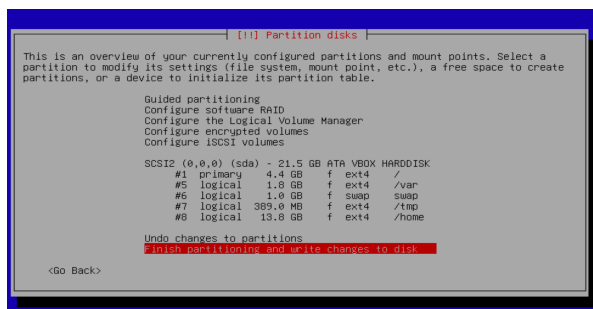


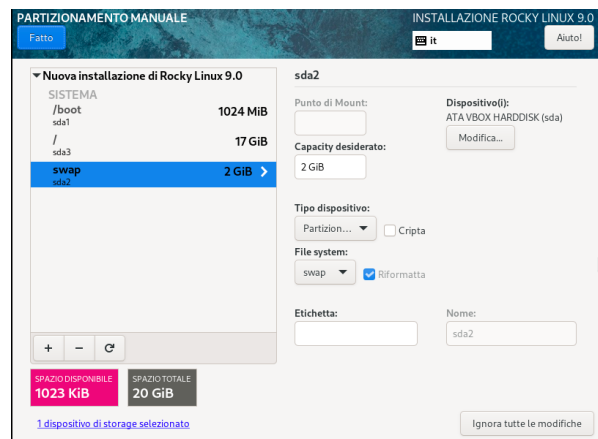
VM exercises

Step 1 - installazione e configurazione VM

1. Scaricare le immagini ISO dei due sistemi: Debian e Rocky Linux 9
2. Creare le due VM con le immagini ISO scaricate (N.B. flaggare il box "installazione non supervisionata" di modo da poter avere il controllo e non usare l'interfaccia grafica)
3. Installazione SO:
 - a. Per creare più partizioni oltre a / durante il setup di installazione → guided
- use entire disk → Selezionare disco → Separate /home, /var and /tmp partitions (Debian)



```
root@vbox:/home/marboccu# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   20G  0 disk
├─sda1       8:1    0    4.1G  0 part /
├─sda2       8:2    0     1K  0 part
├─sda5       8:5    0    1.7G  0 part /var
├─sda6       8:6    0   976M  0 part [SWAP]
├─sda7       8:7    0   371M  0 part /tmp
└─sda8       8:8    0   12.9G  0 part /home
sr0         11:0    1   3.7G  0 rom
```



```
root@vbox:/# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   20G  0 disk
├─sda1       8:1    0     1G  0 part /boot
├─sda2       8:2    0     2G  0 part [SWAP]
├─sda3       8:3    0    17G  0 part /
└─sr0       11:0    1  1024M  0 rom
```

4. Network → Per far sì che le due VM comunichino tra loro, nelle impostazioni di ciascuna VM su virtualbox, l'adapter 1 deve essere Internal Network (N.B. non c'è internet access):

- a. Debian: verificare il nome dell'interfaccia con `ip addr` e poi aggiungere manualmente la configurazione dentro `/etc/network/interfaces` → poi `sudo systemctl restart networking`

```
auto enp0s3
iface enp0s3 inet static
address 192.168.56.101
netmask 255.255.255.0
gateway 192.168.56.1
```

- b. Rocky: verifica il nome dell'interfaccia con `ip addr` e poi modificare il file di configurazione di rete dentro `/etc/sysconfig/network-script/<interfaccia>` oppure usare il comando `nmcli` per configurare la rete (quello che ho fatto io)

```
nmcli connection show
nmcli connection modify enp0s3 ipv4.address 192.168.56.101
nmcli connection modify enp0s3 ipv4.gateway 192.168.56.1
nmcli connection modify enp0s3 ipv4.method manual
nmcli connection modify enp0s3 autoconnect yes
nmcli connection down enp0s3 && nmcli connection up enp0s3
```

```
root@vbox:/home/marboocu# ping 192.168.56.102 -c 5
PING 192.168.56.102 (192.168.56.102) 56(84) bytes of data:
54 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=0.737 ms
54 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=2.07 ms
54 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=0.949 ms
54 bytes from 192.168.56.102: icmp_seq=4 ttl=64 time=1.23 ms
54 bytes from 192.168.56.102: icmp_seq=5 ttl=64 time=0.985 ms

--- 192.168.56.102 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 0.737/1.193/2.069/0.464 ms
```

Debian pinging Rocky

```
root@vbox:/# ping 192.168.56.101 -c 5
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data:
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=1.35 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=1.56 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 192.168.56.101: icmp_seq=4 ttl=64 time=1.07 ms
64 bytes from 192.168.56.101: icmp_seq=5 ttl=64 time=1.38 ms

--- 192.168.56.101 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4022ms
rtt min/avg/max/mdev = 1.073/1.310/1.555/0.165 ms
```

Rocky pinging Debian



MOLTO PIU EASY → Se assegni a entrambe le VMs come scheda di rete **Bridge adapter**, automaticamente il DHCP assegnerà loro un indirizzo Ipv4 dinamico nella stessa rete per far si che possano pingarsi, ma attiva anche la connessione verso internet, per poter scaricare i pacchetti.

5. Installazione di Web Server Apache su Rocky:

- a. `yum update` per aggiornare il packet manager
- b. `yum install httpd -y` per installare Apache Web Server
- c. Per aggiungere il testo "Hello, DevOps Tribe!" possiamo usare il comando `tee` che pipato legge l'input standard e scrive l'output sia su stdout sia nel file specificato, nel nostro caso index.html → `echo "Hello, DevOps Tribe!" | cat tee /var/www/html/index.html`
- d. Per verificare che Apache sia Up&Running → `sudo systemctl status httpd`
- e. I firewall su Rocky potrebbe bloccare il traffico HTTP (porta 80) impedendo così a Debian di raggiungere il server quindi:
 - a. Controllare regole x vedere le porte aperte → `sudo firewall-cmd --list-all`
 - b. Abilitare traffico HTTP → `sudo firewall-cmd --add-service=http`
 - c. Reload → `sudo firewall-cmd --reload`
- f. Test da Rocky stesso → `curl http://localhost`

6. Visualizzazione Web Page su Debian:

- a. Per chiamare Apache da debian dobbiamo usare curl quindi prima di tutto installarlo con `apt-get install curl`
- b. Chiamare Apache → `curl <IPRocky>`

```
root@vbox:/home/marboccu# curl http://192.168.1.46
Hello, DevOps Tribe!
```

Step 2 - Password e Utenti

1. Modificare il file `/etc/login.defs`, in particolare `PASS_MAX_DAYS` e `PASS_WARN_AGE` come nell'immagine di seguito:

```
#  
PASS_MAX_DAYS 90  
PASS_MIN_DAYS 0  
PASS_WARN_AGE 5
```

2. Creare 5 utenti con il comando `sudo adduser <username>`

```
root@vbox:/home/marboccu# cut -d: -f1 /etc/passwd | tail -n 5  
marboccu  
luca  
franci  
lore  
emma
```

3. Per verificare che la policy delle password è stata correttamente applicata all'utente è possibile con `chage -l <username>`

```
root@vbox:/home/marboccu# chage -l luca  
Last password change           : Oct 14, 2024  
Password expires               : Jan 12, 2025  
Password inactive              : never  
Account expires                : never  
Minimum number of days between password change : 0  
Maximum number of days between password change : 90  
Number of days of warning before password expires : 5  
root@vbox:/home/marboccu#
```

Step 3 - Autenticazione SSH

1. Generare la coppia di chiavi RSA sul client (Debian) usando il comando `ssh-keygen -t rsa`

```
marboccu@vbox:~$ ssh-keygen -t rsa  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/marboccu/.ssh/id_rsa):  
Created directory '/home/marboccu/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/marboccu/.ssh/id_rsa  
Your public key has been saved in /home/marboccu/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:KH2RMTrI2LY/JVwsX1dWAjm4Tkz6m6lQ/00oyC03HTs marboccu@vbox  
The key's randomart image is:
```

2. Ora bisogna copiare la public key sul server usando l'utility `ssh-copy-id` e specificando lo username sul quale sono salvate le chiavi SSH (nel mio caso `marboccu`) e l'IP dell'host remoto (server Apache) → `ssh-copy-id marboccu@192.168.1.46`
3. Verificare che la copia sia riuscita provando a autenticarsi tramite ssh sul client → `ssh marboccu@192.168.1.46` e come si vede dallo screen sotto, la connessione è stata effettuata e sono effettivamente dentro il server:

```
marboccu@vbox:~$ ssh marboccu@192.168.1.46
Last login: Mon Oct 14 11:14:50 2024 from 192.168.1.188
[marboccu@localhost ~]$ curl http://localhost
Hello, DevOps Tribe!
[marboccu@localhost ~]$ exit
logout
Connection to 192.168.1.46 closed.
marboccu@vbox:~$
```

Step 4 - Permessi, gruppi

1. Creare il file con `touch foo.log` e assegnare i permessi solo all'owner di lettura, scrittura ed esecuzione con `chmod 700 foo.log`

```
marboccu@vbox:~$ touch foo.log
marboccu@vbox:~$ chmod 700 foo.log
marboccu@vbox:~$ ls -l foo.log
-rwx----- 1 marboccu marboccu 0 Oct 14 06:11 foo.log
marboccu@vbox:~$
```

2. Creare il gruppo foobar con `sudo addgroup foobar` e assegnarlo come gruppo proprietario di `foo.log` con `sudo chown marboccu:foobar foo.log`

```
marboccu@vbox:~$ sudo chown marboccu:foobar foo.log
marboccu@vbox:~$ ls -l
total 0
-rwx----- 1 marboccu foobar 0 Oct 14 06:11 foo.log
marboccu@vbox:~$
```

Step 5 - Esercizi pdf

- Determinare il numero di files nella directory `/bin` la cui prima lettera è "c":
 - `ls /bin/c* | wc -l`

- Creare un file contenente i nomi dei primi 7 files della directory /etc:
 - `ls /etc | head -n 7 > first_etc.txt`
- Determinare il numero di file della current dir nel cui nome compare "string"
 - `ls /*string* | wc -l`
- Creare un file contenente una lista col nome di 10 comandi di /bin ordinati secondo il momento dell'ultimo accesso
 - `ls -u /bin | head -n 11 > last_cmds.txt`
- Creare un file contenente i nomi dei primi 7 files e gli ultimi 6 files (in ordine alfabetico) della directory /etc
 - `(ls /etc | head -n 7; ls /etc | tail -n 6) > files.txt`
- Creare un file contenente una lista coi nomi di 8 files in /usr/sbin ordinati secondo il momento dell'ultima modifica
 - `ls -t /usr/sbin | head -n 8 > last_edit.txt`

Riporto anche gli esercizi che nel pdf avevano già la soluzione per completezza:

- Si stampi l'insieme dei metadati per diversi file (es. \$HOME, /dev/sda1)
 - `stat <file>`
- Si crei un file nuovo di contenuto nullo e si individui un modo per importare il timestamp al 1/1/1970, ore 00:00
 - `touch -t 197001010000.00 file.txt`
- Si crei una gerarchia di sub-directory, dir3 dentro dir2 dentro dir1
 - `mkdir -p dir1/dir2/dir3`
- Si individui un modo per forzare la cancellazione di file in maniera batch
 - `rm -f file.txt`
- Cosa fa in più l'opzione -a di cp rispetto a --preserve=all?
 - l'opzione -a invoca sia l'opzione --preserve=all ma anche l'opzione -d e -R
- Si applichi un metodo per stampare il file mkfs.trace al contrario

- `tac mkfs.trace`
- Si visualizzino in formato lungo i 5 file più recenti nella directory attuale
 - `ls -lt . | head -n 6`
- Si produca un file contenente gli username e le shell da essi utilizzate
 - `cut -f1,7 -d: /etc/passwd > users_shells.txt`
- Si ordini il file /etc/passwd per ordinamento numerico crescente e per campo GID
 - `sort -n -k 4 -t : /etc/passwd`
- Nell'ipotesi che un file di configurazione verifichi il pattern di shell *.log, si individuino tutti i file di log nel sistema
 - `find / -name "*.log"`
- Si individuino tutti i file HTML nel sistema
 - `find / -iregex "^.*\\.html$"`
- Si individuino tutti i file più grandi di 1MB presenti nell'intero sistema
 - `find / -size +1M`
- Si produca un elenco di tutti i file nel sistema con relativa dimensione. Si ordini l'elenco secondo tale dimensione e si estraiga la top ten dei file più grandi
 - `find / printf "%p $s\n" | sort -nrk 2 | head -n 10`
- Si individuino tutti i file di estensione .trace o .strace e li si cancellino forzatamente
 - `find / -regex '^.*\\.s?trace$' -exec rm -f '{}' \;`