Che cos'è una Virtual Machine?

Una Virtual Machine (VM) è un software che crea un ambiente virtuale all'interno di un sistema operativo ospite in modo da consentire l'esecuzione di un sistema operativo ospite separato. In altre parole, una VM è un'istanza isolata di un sistema operativo completo, compresi i file di sistema, le applicazioni e le librerie, che viene eseguita all'interno di un altro sistema operativo ospite.

Questo ambiente, chiamato "macchina virtuale", viene creato dal software di virtualizzazione intercettando l'accesso a determinati componenti hardware e determinate funzionalità. Il computer fisico viene solitamente chiamato "host", mentre la macchina virtuale viene spesso chiamata "guest". La maggior parte del codice guest viene eseguito senza modifiche direttamente sul computer host e il sistema operativo guest "pensa" di essere in esecuzione su una macchina reale.

Le VM consentono ai programmatori di testare il software in un ambiente controllato e separato, senza dover installare un sistema operativo separato su un hardware fisico dedicato. Le VM sono utilizzate anche per la virtualizzazione del server, in cui una singola macchina fisica viene suddivisa in più VM per aumentare l'utilizzo delle risorse del server.

Le VM possono essere utilizzate anche per l'isolamento e la sicurezza, in cui il software viene eseguito all'interno di una VM che limita l'accesso alle risorse del sistema ospite. In questo modo, se il software nella VM viene compromesso, il sistema operativo ospitante rimane protetto.

Ci sono diversi scenari che rendono la virtualizzazione allettante:

- Supporto del sistema operativo. Con un virtualizzatore come VirtualBox, è possibile eseguire software scritto per un sistema operativo su un altro (ad esempio, software Windows su Linux) senza dover riavviare.
- Consolidamento dell'infrastruttura. Poiché le prestazioni complete dei computer di oggi raramente sono necessarie a tempo pieno, invece di far funzionare molti di questi computer fisici, è possibile "imballare" molte macchine virtuali su pochi host potenti e bilanciare i carichi tra di loro. Ciò può risparmiare molti costi hardware: ad esempio, consolidando molti server in pochi.
- Test e ripristino in caso di disastro. Soprattutto con l'uso degli snapshot, è possibile sperimentare con un ambiente informatico eseguendolo come macchina virtuale. Se qualcosa va storto, è possibile tornare facilmente a uno snapshot precedente e evitare la necessità di frequenti backup e ripristini.

Qual è lo scopo di una Virtual Machine?

Lo scopo principale di una Virtual Machine (VM) è quello di creare un ambiente virtuale che emuli un intero sistema informatico, incluso il sistema operativo, le applicazioni e le librerie, all'interno di un altro sistema operativo ospite. Ciò consente di eseguire più sistemi operativi su un singolo sistema fisico, consentendo l'isolamento e la gestione delle risorse del sistema, come CPU, RAM, storage e rete.

Ci sono molte ragioni per utilizzare una VM, tra cui:

- Testing: le VM consentono di testare il software in un ambiente controllato, senza dover installare un sistema operativo separato su un hardware dedicato.
- Sviluppo: le VM consentono di creare ambienti di sviluppo separati e isolati per diverse applicazioni e versioni del software.
- Consolidamento dei server: le VM possono consolidare più server fisici in un singolo sistema fisico, migliorando l'utilizzo delle risorse e riducendo i costi di gestione.
- Migrazione: le VM possono essere utilizzate per migrare il software da un sistema operativo a un altro, senza dover reinstallare o configurare nuovamente il software.
- Sicurezza: le VM possono essere utilizzate per l'isolamento e la sicurezza, in cui il software viene eseguito all'interno di una VM che limita l'accesso alle risorse del sistema ospite.

In sintesi, una VM consente di creare ambienti di lavoro isolati e sicuri, migliorando l'efficienza e la gestione delle risorse del sistema.

Quali sono le differenze base tra CentOS e Debian come sistemi operativi?

CentOS e Debian sono entrambi sistemi operativi basati su Linux, ma ci sono alcune differenze fondamentali tra di loro:

- Origine: CentOS è basato su Red Hat Enterprise Linux (RHEL), mentre Debian è stato sviluppato indipendentemente.
- Ciclo di rilascio: CentOS ha un ciclo di rilascio molto più lento rispetto a Debian. CentOS ha una politica di supporto a lungo termine (LTS) e ogni versione è supportata per 10 anni. Debian ha un ciclo di rilascio più rapido e ogni versione ha un supporto di circa 5 anni.
- Gestore di pacchetti: CentOS utilizza il gestore di pacchetti YUM (Yellowdog Updater Modified), mentre Debian utilizza il gestore di pacchetti APT (Advanced Package Tool).
- Comunità di sviluppo: Debian è sviluppato da una comunità di volontari, mentre CentOS è principalmente sviluppato da Red Hat.
- Supporto hardware: CentOS ha un forte supporto per hardware server di fascia alta, mentre Debian ha un supporto più ampio per hardware generico.
- Installazione e configurazione: L'installazione di Debian è generalmente considerata più complessa rispetto a CentOS. Tuttavia, Debian offre una maggiore flessibilità nella configurazione del sistema operativo rispetto a CentOS.
- Stabilità: Entrambi i sistemi operativi sono noti per la loro stabilità e affidabilità, ma CentOS è generalmente considerato il sistema operativo più stabile e affidabile, in particolare per l'uso su server aziendali critici.

In sintesi, entrambi i sistemi operativi sono robusti e affidabili, ma CentOS è generalmente considerato la scelta migliore per le aziende che cercano una piattaforma stabile e sicura per i loro server, mentre Debian è più flessibile e adatto per gli utenti più avanzati che hanno bisogno di un alto grado di personalizzazione e controllo del loro sistema operativo.

Che cosa é un SSH service?

Un SSH (Secure Shell) service è un servizio di rete che permette di stabilire una connessione sicura e cifrata tra due dispositivi tramite un protocollo di comunicazione chiamato SSH.

In pratica, SSH consente a un utente di connettersi ad un server remoto in modo sicuro, autenticandosi con una coppia di chiavi (una pubblica e una privata), e di accedere al prompt della shell del server per eseguire comandi come se fosse sul dispositivo locale.

L'utilizzo di SSH è particolarmente importante per la sicurezza delle connessioni remote, in quanto crittografa il traffico dati scambiato tra i due dispositivi, rendendo impossibile per un malintenzionato intercettare e decodificare i dati trasmessi. Inoltre, SSH offre anche una varietà di funzionalità utili, come la copia sicura di file e la gestione delle chiavi di accesso.

Controlla che il SSH service é avviato.

root@misidori42:/home/misidori# sudo service ssh status
● ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
Active: ***active (running)*** *since Mon 2023-02-27 04:33:07 CET; 44min ago*
Docs: man:sshd(8)
man:sshd_config(5)
Process: 33866 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
Main PID: 33867 (sshd)
Tasks: 1 (limit: 1125)
Memory: 2.4M
CPU: 45ms
CGroup: /system.slice/ssh.service
└─33867 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Feb 27 04:33:07 misidori42 systemd[1]: Starting OpenBSD Secure Shell server...
Feb 27 04:33:07 misidori42 sshd[33867]: Server listening on 0.0.0.0 port 4242.
Feb 27 04:33:07 misidori42 sshd[33867]: Server listening on :: port 4242.
Feb 27 04:33:07 misidori42 systemd[1]: Started OpenBSD Secure Shell server.

Che cos'é sudo?
sudo è un comando utilizzato nei sistemi operativi basati su Unix come Linux e macOS, che consente agli utenti di eseguire comandi con privilegi amministrativi (superuser) senza dover accedere come root. La parola "sudo" significa "superuser do", ovvero "esegui come superutente".

In pratica, sudo consente di ottenere temporaneamente i permessi di amministratore per eseguire specifici comandi o attività che richiedono privilegi elevati, come l'installazione di pacchetti software o la modifica di file di configurazione del sistema.

L'utilizzo di sudo è utile per motivi di sicurezza, poiché consente di limitare l'accesso agli account di amministratore solo ai momenti in cui è strettamente necessario. In questo modo si riduce il rischio di attacchi informatici e di danni causati da errori umani.

Che cos'é un firewall?
Un firewall è un software o un hardware che agisce come una barriera di sicurezza tra una rete di computer e Internet. Il suo scopo principale è quello di proteggere la rete bloccando l'accesso non autorizzato da parte di utenti esterni o di programmi dannosi.

Il firewall può essere configurato per bloccare il traffico in base a determinati criteri, come l'indirizzo IP di origine, il tipo di protocollo o la porta utilizzata per la connessione. In questo modo, il firewall può prevenire attacchi informatici come hacking, malware e phishing, proteggendo la rete e i dati sensibili che contiene.

Il firewall può anche essere configurato per limitare l'accesso a determinati siti web o applicazioni da parte degli utenti della rete, aiutando così a controllare l'uso della rete e migliorare la produttività dei dipendenti. In generale, il firewall è un componente importante della sicurezza informatica e della protezione della privacy online.

Che cos'é il UFW firewall?

UFW (Uncomplicated Firewall) è un firewall basato su IPtables che fornisce un'interfaccia semplificata per configurare le regole di filtraggio dei pacchetti su un sistema Linux.

UFW viene installato di default in alcune distribuzioni Linux, come ad esempio Ubuntu, e può essere gestito tramite il comando "ufw" da riga di comando o tramite GUI.

Le regole di UFW consentono di specificare quali porte e protocolli di rete sono aperti o chiusi, e di impostare le politiche di default per le connessioni in entrata e in uscita. È possibile anche configurare regole più avanzate per permettere o bloccare il traffico di rete da o verso specifici indirizzi IP o sottoreti.

UFW è utile per migliorare la sicurezza del sistema, in quanto permette di bloccare l'accesso non autorizzato alle porte di rete e di limitare il traffico di rete entrante e uscente. Tuttavia, è importante ricordare che UFW è solo uno strumento di sicurezza tra molti altri che dovrebbero essere utilizzati per proteggere un sistema Linux.

A cosa serve il comando ufw status?
Il comando ufw status è un comando utilizzato per verificare lo stato del firewall UFW (Uncomplicated Firewall) su un sistema Linux. UFW è un frontend semplificato per iptables e consente agli amministratori di sistema di configurare rapidamente e facilmente il firewall di sistema.

Il comando ufw status restituisce lo stato corrente del firewall UFW, inclusi i profili attivi e le regole configurate.

Controlla che il UFW service é avviato.
root@misidori42:~# /usr/sbin/ufw status
Status: active

To	*Action*	*From*
--	*------*	*----*
4242	*ALLOW*	*Anywhere*
4242 (v6)	*ALLOW*	*Anywhere (v6)*

In questo esempio, il firewall UFW è attivo e ci sono alcune regole configurate che consentono il traffico attraverso la porta 4242.

Il comando ufw status è utile per verificare lo stato del firewall UFW e per identificare eventuali problemi di configurazione o errori nelle regole del firewall.

Per aggiungere una porta:
sudo ufw allow 8080
Esempio:
root@misidori42:/home/misidori# sudo ufw allow 8080
Rule added
Rule added (v6)
root@misidori42:/home/misidori# sudo ufw status
Status: active

To	Action	From
4242	ALLOW	Anywhere
8080	ALLOW	Anywhere
4242 (v6)	ALLOW	Anywhere (v6)
8080 (v6)	ALLOW	Anywhere (v6)

Per chiudere una porta:
sudo ufw deny 8080
Esempio:
root@misidori42:/home/misidori# sudo ufw deny 8080
Rule updated
Rule updated (v6)
root@misidori42:/home/misidori# sudo ufw status
Status: active

To	Action	From
4242	ALLOW	Anywhere
8080	DENY	Anywhere
4242 (v6)	ALLOW	Anywhere (v6)
8080 (v6)	DENY	Anywhere (v6)

Per eliminare una porta:
sudo ufw delete deny 8080
root@misidori42:/home/misidori# sudo ufw status
Status: active

To	Action	From
4242	ALLOW	Anywhere

4242 (v6) ALLOW Anywhere (v6)

Che cos'é l'hostname?
L'hostname è il nome assegnato a un computer o ad un dispositivo collegato in una rete.

In altre parole, l'hostname identifica univocamente un dispositivo nella rete e consente di accedervi tramite la rete utilizzando il suo nome invece dell'indirizzo IP. Ad esempio, se l'hostname di un computer è "mio_pc", altri dispositivi sulla rete possono accedere a quel computer utilizzando l'indirizzo "mio_pc" invece dell'indirizzo IP numerico.

L'hostname è un parametro di configurazione del sistema operativo che può essere impostato dall'utente.

Comandi da sapere:

Per verificare il sistema operativo montato:
root@misidori42:/home/misidori# uname -a
Linux misidori42 5.10.0-21-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64 GNU/Linux
oppure
root@misidori42:/home/misidori# uname -v
#1 SMP Debian 5.10.162-1 (2023-01-21)
oppure
root@misidori42:/home/misidori# head -n 2 /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"

Creare un nuovo utente:
sudo adduser "name_user"
Esempio:
root@misidori42:/home/misidori# sudo adduser scabbol
Adding user `scabbol' ...
Adding new group `scabbol' (1002) ...
Adding new user `scabbol' (1001) with group `scabbol' ...
Creating home directory `/home/scabbol' ...
Copying files from `/etc/skel' ...
*New password: ****
*Retype new password: ****
passwd: password updated successfully
Changing the user information for scabbol

Vedere la lista degli utenti:
cat /etc/passwd

Creare un nuovo gruppo:
sudo addgroup "group_name"

Aggiungere un utente in un gruppo:
sudo adduser name_user group_name
Esempio:
root@misidori42:/home/misidori# sudo adduser scabbol user42
Adding user `scabbol' to group `user42' ...
Adding user scabbol to group user42
Done.

Vedere la lista di utenti di un gruppo:
getent group group_name
Esempio:
root@misidori42:/home/misidori# getent group user42

user42:x:1001:misidori,scabbol

Eliminare un utente:
sudo deluser user_name
Esempio:
root@misidori42:/home/misidori# sudo deluser scabbol
Removing user `scabbol' ...
Warning: group `scabbol' has no more members.
Done.
Lanciando pero' il comando
root@misidori42:/home/misidori# sudo groupdel scabbol
groupdel: group 'scabbol' does not exist
scabbol non esiste piú neanche come gruppo.

Eliminare un utente da un gruppo:
sudo deluser name_user group_name
Esempio:
root@misidori42:/home/misidori# sudo deluser scabbol user42
Removing user `scabbol' from group `user42' ...
Done.

Eliminare un gruppo:
sudo groupdel group_name
Esempio:
root@misidori42:/home/misidori# sudo groupdel user42

Vedere il nome dell'host:
root@misidori42:/home/misidori# hostname
misidori42

Modificare il nome dell'host:
sudo nano /etc/hostname
sudo nano /etc/hosts
e riavviare la macchina virtuale. Riloggando si noterá il cambio del nome dell'host.

Cambiare password di un utente:
Esempio:
sudo passwd root

Per connettersi via SSH dalla macchina ospitante alla macchina virtuale ospite:
ssh misidori@localhost -p 4242
verrá chiesta la password dell'utente con cui stiamo cercando di loggarci sulla virtual machine. Una volta che la password verrá immessa, il login verrá effettuato e il nome dell'utente apparirá verde, a significare che la connessione é stata stabilita correttamente.

monitoring.sh

```
#!/bin/bash

# ARCH
arch=$(uname -a)

# CPU PHYSICAL
cpuf=$(grep "physical id" /proc/cpuinfo | wc -l)

# CPU VIRTUAL
cpuv=$(grep "processor" /proc/cpuinfo | wc -l)

# RAM
ram_total=$(free --mega | awk '$1 == "Mem:" {print $2}')
ram_use=$(free --mega | awk '$1 == "Mem:" {print $3}')
ram_percent=$(free --mega | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')

# DISK
disk_total=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_t += $2} END {printf
("%.1fGb\n"), disk_t/1024}')
disk_use=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} END {print disk_u}')
disk_percent=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} { disk_t += $2} END
{printf("%d"), disk_u/disk_t*100}')

# CPU LOAD
cpul=$(vmstat 1 2 | tail -1 | awk '{printf $15}')
cpu_op=$(expr 100 - $cpul)
cpu_fin=$(printf "%.1f" %cpu_op)

# LAST BOOT
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')

# LVM USE
lvmu=$(if [ $(lsblk | grep "lvm" | wc -l) -gt 0 ]; then echo yes; else echo no; fi)

# TCP CONNECTIONS
tcpc=$(ss -ta | grep ESTAB | wc -l)

# USER LOG
ulog=$(users | wc -w)

# NETWORK
ip=$(hostname -I)
mac=$(ip link | grep "link/ether" | awk '{print $2}')

# SUDO
cmnd=$(journalctl _COMM=sudo | grep COMMAND | wc -l)
```

wall "

#Architecture: $arch
#CPU physical: $cpuf
#vCPU: $cpuv
#Memory Usage: $ram_use/${ram_total}MB ($ram_percent%)
#Disk Usage: $disk_use/${disk_total} ($disk_percent%)
#CPU load: $cpu_fin%
#Last boot: $lb
#LVM use: $lvmu
#Connections TCP: $tcpc ESTABLISHED
#User log: $ulog
#Network: IP $ip ($mac)
#Sudo: $cmnd cmd"

In output si avrá:

Broadcast message from root@misidori42 (somewhere) (Mon Feb 27 03:00:02 2023):

#Architecture: Linux misidori42 5.10.0-21-amd64 #1 SMP Debian 5.10.162-1 (2023-01-21) x86_64 GNU/Linux
#CPU physical: 1
#vCPU: 1
#Memory Usage: 171/1024MB (16.70%)
#Disk Usage: 1997/25.7Gb (7%)
#CPU load: 0.0%
#Last boot: 2023-02-26 22:19
#LVM use: yes
#Connections TCP: 2 ESTABLISHED
#User log: 3
#Network: IP 10.0.2.15 (08:00:27:f0:28:67)
#Sudo: 108 cmd

Che cos'è il comando awk?

Awk è un potente strumento di manipolazione di testo che viene utilizzato principalmente per l'estrazione e l'elaborazione di dati strutturati in file di testo.

Awk legge il file di input riga per riga e applica un insieme di regole specificate dall'utente, chiamate "pattern-action", per elaborare i dati. I pattern sono espressioni regolari che corrispondono a un certo tipo di righe nel file di input, mentre l'action definisce l'azione da eseguire su tali righe. L'azione può includere l'elaborazione dei dati, la stampa di output e la manipolazione delle variabili di Awk.

Che cosa fa il simbolo $ in bash?

Il simbolo "$?" è una variabile shell speciale in Bash (e in altre shell Unix-like) che contiene il codice di uscita dell'ultimo comando eseguito.

Che cosa fa il comando grep?

grep è un comando di ricerca di testo utilizzato nei sistemi operativi Unix e Unix-like, incluso Linux. Il nome "grep" è l'abbreviazione di "global regular expression print". Il comando grep cerca all'interno di uno o più file (o dell'output di un altro comando) per un pattern specificato dall'utente e stampa le righe che contengono corrispondenze con il pattern.
L'opzione -v del comando grep indica di selezionare tutte le righe che NON contengono il pattern cercato.

Vediamo il codice passo per passo:

#!/bin/bash
questa è la dichiarazione del tipo di shell che verrà utilizzata per eseguire lo script.

ARCH
arch=$(uname -a)
Questa variabile contiene la stringa di output del comando "uname -a" che restituisce informazioni sull'architettura del sistema operativo.

CPU PHYSICAL
cpuf=$(grep "physical id" /proc/cpuinfo | wc -l)
questa variabile contiene il numero di CPU fisiche presenti nel sistema operativo. Per ottenere questa informazione, lo script cerca la stringa "physical id" nel file /proc/cpuinfo e conta il numero di occorrenze trovate con il comando "wc -l".

CPU VIRTUAL
cpuv=$(grep "processor" /proc/cpuinfo | wc -l)
questa variabile contiene il numero di CPU virtuali presenti nel sistema operativo. Per ottenere questa informazione, lo script cerca la stringa "processor" nel file /proc/cpuinfo e conta il numero di occorrenze trovate con il comando "wc -l".

RAM
ram_total=$(free --mega | awk '$1 == "Mem:" {print $2}')
ram_use=$(free --mega | awk '$1 == "Mem:" {print $3}')
*ram_percent=$(free --mega | awk '$1 == "Mem:" {printf("%.2f"), $3/$2*100}')*
questa sezione raccoglie informazioni sulla memoria RAM.
La variabile "ram_total" contiene la quantità totale di RAM installata nel sistema, ottenuta dal comando "free --mega" e filtrata con il comando "awk".
Il comando *free --mega* restituisce informazioni sulla memoria del sistema. In particolare, l'opzione --mega indica di mostrare i risultati in Megabyte invece di Kilobyte.
La variabile "ram_use" contiene la quantità di RAM attualmente utilizzata. La variabile "ram_percent" contiene la percentuale di RAM utilizzata.

DISK
disk_total=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_t += $2} END {printf ("%.1fGb\n"), disk_t/1024}')
disk_use=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} END {print disk_u}')
*disk_percent=$(df -m | grep "/dev/" | grep -v "/boot" | awk '{disk_u += $3} { disk_t += $2} END {printf("%d"), disk_u/disk_t*100}')*

Il comando *df -m* viene utilizzato per estrarre la quantità totale di spazio su disco disponibile e la quantità di spazio su disco utilizzata. Queste informazioni vengono quindi utilizzate per calcolare la percentuale di spazio su disco utilizzato.
questa sezione raccoglie informazioni sulle unità di disco presenti nel sistema. La variabile "disk_total" contiene la quantità totale di spazio disponibile sui dischi, ottenuta dal comando "df -m" e filtrata con il comando "awk".
END viene utilizzato con il comando awk per eseguire un'operazione (ad esempio, un calcolo o una stampa di output) dopo che awk ha analizzato completamente il file di input. In particolare, il comando *awk '{disk_t += $2} END {printf("%.1fGb\n"), disk_t/1024}'* viene utilizzato per sommare le dimensioni dei file system individuati dal comando df -m e convertire il risultato da megabyte a gigabyte.
La variabile "disk_use" contiene la quantità di spazio attualmente utilizzata. La variabile "disk_percent" contiene la percentuale di spazio utilizzato.

CPU LOAD
cpul=$(vmstat 1 2 | tail -1 | awk '{printf $15}')
cpu_op=$(expr 100 - $cpul)
cpu_fin=$(printf "%.1f" %cpu_op)
questa sezione raccoglie informazioni sulla cpu. La variabile "cpu_fin" contiene la percentuale di utilizzo della CPU approssimata a una cifra decimale.
Il comando "vmstat 1 2" viene eseguito per leggere le statistiche di memoria 2 volte a distanza di un secondo.
Il comando "tail -1" estrae l'ultima riga dell'output di vmstat, ovvero la riga che contiene le statistiche di memoria aggiornate.
Il comando *expr* esegue l'aritmetica di shell. In questo caso, l'operazione eseguita è la sottrazione tra il valore 100 e il contenuto della variabile *$cpul*.

LAST BOOT
lb=$(who -b | awk '$1 == "system" {print $3 " " $4}')
Il comando who mostra gli utenti attualmente connessi al sistema e l'ultima volta che hanno effettuato il login.
questa variabile contiene la data e l'ora dell'ultimo avvio del sistema. Per ottenere questa informazione, lo script cerca la stringa "system" nel file /var/log/wtmp e utilizza il comando "awk" per estrarre la data e l'ora.
L'opzione *-b* del comando who mostra l'ultima volta che il sistema è stato avviato, ovvero la data e l'ora di boot del sistema.

LVM USE
lvmu=$(if [$(lsblk | grep "lvm" | wc -l) -gt 0]; then echo yes; else echo no; fi)
questa variabile indica se il sistema utilizza la gestione logica dei volumi (LVM) per la gestione dello spazio su disco.
Se il sistema utilizza LVM, la variabile è impostata su "yes", altrimenti su "no". Per determinare se il sistema utilizza LVM, lo script utilizza il comando "*lsblk*" (che elenca tutte le unità di archiviazione presenti nel sistema, come dischi rigidi, partizioni, dispositivi USB, ecc.), quindi cerca la stringa "lvm" e conta il numero di occorrenze trovate.

TCP CONNECTIONS
tcpc=$(ss -ta | grep ESTAB | wc -l)

questa variabile contiene il numero di connessioni TCP in stato ESTABLISHED nel sistema. Per ottenere questa informazione, lo script utilizza il comando "ss" per visualizzare le informazioni sulle connessioni di rete, quindi cerca la stringa "ESTAB" e conta il numero di occorrenze trovate.
ss -ta è un comando che utilizza il comando ss per mostrare tutte le connessioni TCP attive, sia in ingresso che in uscita, inclusi i socket di ascolto. La flag -t specifica che devono essere mostrate solo le connessioni TCP, mentre la flag -a specifica di mostrare anche le connessioni in ascolto.

USER LOG
ulog=$(users | wc -w)
questa variabile contiene il numero di utenti connessi al sistema. Per ottenere questa informazione, lo script utilizza il comando "users" per visualizzare i nomi degli utenti connessi, quindi conta il numero di parole trovate con il comando "wc -w".

NETWORK
ip=$(hostname -I)
mac=$(ip link | grep "link/ether" | awk '{print $2}')
questa sezione raccoglie informazioni sulla rete. La variabile "ip" contiene l'indirizzo IP del sistema, ottenuto dal comando "hostname -I". La variabile "mac" contiene l'indirizzo MAC dell'interfaccia di rete, ottenuto dal comando "ip link".

SUDO
cmnd=$(journalctl _COMM=sudo | grep COMMAND | wc -l)
questa variabile contiene il numero di comandi eseguiti con il comando "sudo" nel sistema. Questo comando è composto da tre parti:

1. *journalctl _COMM=sudo*: questo comando esegue una ricerca nel registro di sistema (journalctl) per tutte le voci in cui il comando sudo è stato eseguito. _COMM è un filtro che limita la ricerca solo alle voci contenenti il comando specificato.
2. *grep COMMAND*: questo comando cerca solo le voci che contengono la parola "COMMAND". In questo modo, viene identificata la riga corrispondente all'esecuzione del comando sudo.
3. *wc -l*: infine, il comando wc conta il numero di linee che corrispondono alla ricerca precedente.

Quindi, la variabile *cmnd* contiene il numero di comandi eseguiti con *sudo* nel registro di sistema.

"wall": questo comando visualizza il messaggio a video utilizzando il comando "wall". Il messaggio contiene le informazioni raccolte dalle variabili precedenti, formattate in modo da essere facilmente leggibili.

Il comando wall in Bash è un'utilità che consente di inviare un messaggio a tutti gli utenti collegati ad un sistema Unix o Linux. La parola "wall" è l'abbreviazione di "write all" e il comando permette di scrivere un messaggio su tutti i terminali degli utenti collegati.

Il modo più semplice per utilizzare il comando wall è il seguente:
wall "Messaggio da inviare a tutti gli utenti"

Una volta eseguito il comando, il messaggio verrà inviato a tutti gli utenti collegati al sistema. Il messaggio verrà visualizzato sui loro terminali in modo simile a quanto segue:

Broadcast message from user@hostname
Messaggio da inviare a tutti gli utenti

È importante notare che per utilizzare il comando wall è necessario avere i privilegi di amministratore o di superutente.

Il codice è uno script di bash che raccoglie informazioni sul sistema operativo e le visualizza a video tramite il comando "wall".

9-2 Evaluation commands ⌨️
1 ◦ Verify that no graphical interface is in use.

We will use the command ls /usr/bin/*session and it should give the same result as in the screenshot. If anything different appears, a graphical interface is being used.

2 ◦ Check that the UFW service is in use.

sudo ufw status

sudo service ufw status

3 ◦ Check that the SSH service is in use.

sudo service ssh status

4 ◦ Check that you are using the Debian or Centos operating system.

uname -v o uname --kernel-version

5 ◦ Check that your user is within the "sudo" and "user42" groups.

getent group sudo

getent group user42

6 ◦ Create a new user and show that it follows the password policy we have created.

sudo adduser name_user and enter a password that follows the policy.

7 ◦ We create a new group named "evaluating".

sudo addgroup evaluating

8 ◦ We add the new user to the new group.

sudo adduser name_user evaluating

To verify that it has been entered correctly.

9 ◦ Check that the machine's hostname is correct login42.

10 ◦ Modify hostname to replace your login with the evaluator's. In this case, we will replace it with student42.

sudo nano /etc/hostname and replace our login with the new one.

sudo nano /etc/hosts and replace our login with the new one.

Reboot the machine.

Once we have logged in again, we can see how the hostname has been changed correctly.

11 ◦ Check that all partitions are as indicated in the subject.

lsblk

12 ◦ Check that sudo is installed.

which sudo

Using which is not actually a good practice as not all packages are found in the paths where which searches. However, for the evaluation it is better as it is a simple and easy-to-learn command. For better use, we will use the following command:

dpkg -s sudo

13 ◦ Add the new user to the sudo group.

sudo adduser name_user sudo

We check that it is within the group.

14 ◦ Show the application of the rules imposed for sudo by the subject.

15 ◦ Show that the path /var/log/sudo/ exists and contains at least one file, in this we should see a history of the commands used with sudo.

Run a command with sudo and check that the file is updated.

16 ◦ Check that the UFW program is installed on the virtual machine and check that it works correctly.

dpkg -s ufw

sudo service ufw status

17 ◦ List the active rules in UFW, if the bonus part is not done, the rule for port 4242 should only appear.

sudo ufw status numbered

18 ◦ Create a new rule for port 8080. Verify that it has been added to the active rules and then you can delete it.

sudo ufw allow 8080 to create it.

sudo ufw status numbered

To delete the rule, we must use the command. sudo ufw delete num_rule

We check that it has been deleted and we see the number of the next rule that needs to be deleted..

Delete the new rule.

We check that only the required rules in the subject remain.

19 ◦ Check that the ssh service is installed on the virtual machine, that it works correctly, and that it only works on port 4242.

which ssh

sudo service ssh status

20 ◦ Use ssh to log in with the newly created user. Make sure that you cannot use ssh with the root user.

We try to connect over ssh with the root user but we do not have permission.

We connect via ssh with the new user using the command ssh newuser@localhost -p 4242

21 ◦ Modify the runtime of the script from 10 minutes to 1.

We run the following command to modify the crontab file sudo crontab -u root -e

We modify the first parameter, instead of 10 we change it to 1.

22 ◦ Finally, make the script stop running when the server has started, but without modifying the script.

sudo /etc/init.d/cron stop

If we want it to run again:

sudo /etc/init.d/cron start