30/06/23

## File permission and users.

Apri il terminale.

Utilizza il seguente comando per aprire l'editor di testo Nano e creare/modificare il file hi:

**sudo nano ~/hi**

Nell'editor di testo Nano, digita il seguente testo:

**#!/bin/bash**

**echo "Hi! how are you?"**

Questo è uno script bash molto semplice che stampa il messaggio "Hi! how are you?" quando viene eseguito.

Ora, salva le modifiche e chiudi l'editor Nano. Premi Ctrl + X, quindi premi Y per confermare le modifiche e infine premi Invio.

Dopo aver salvato il file, rendilo eseguibile con il comando:

**sudo chmod a+x ~/hi**

Ora puoi eseguire lo script con il comando:

**~/hi**

## Shell environment

Per l'utente "student":

Apri un terminale e digita il seguente comando per modificare il file **.bashrc** dell'utente "student". Se il file non esiste, verrà creato:

**nano ~/.bashrc**

Aggiungi la seguente linea alla fine del file .bashrc:

**export HELLO\_USER="hello student"**

Per tutti gli utenti del sistema:

Per fare in modo che questa variabile sia disponibile per tutti gli utenti del sistema, modifica il file /etc/environment usando il comando sudo:

**sudo nano /etc/environment**

Aggiungi la seguente linea al file:

**HELLO\_ALL\_USERS="hello all users"**

**(per salvare il file ctrl+x e poi y + invio )**

## File system permission and user

Creazione del gruppo "months" (se non esiste):

Se il gruppo "months" non esiste, puoi crearlo utilizzando il comando groupadd:

**sudo groupadd months**

Creazione dell'utente "june" con home directory personalizzata:

Usa il comando useradd per creare l'utente "june" e specificare la directory home personalizzata:

**sudo useradd -m -d /home/months/june -g months June**

-m crea la directory home se non esiste.

-d specifica il percorso della directory home.

-g imposta il gruppo primario dell'utente su "months".

Impostazione dei permessi sulla directory home:

Assicurati che la directory home di "june" sia presente con i permessi corretti:

**sudo mkdir -p /home/months/june**

**sudo chown june:months /home/months/June**

## HTTPD

Installazione di Apache:

Apri il terminale e utilizza il gestore di pacchetti dnf per installare Apache:

**sudo dnf install httpd**

Abilita e avvia Apache:

Dopo l'installazione, attiva il servizio Apache in modo che venga eseguito all'avvio e avvialo:

**sudo systemctl enable httpd**

**sudo systemctl start httpd**

Utilizza il tuo editor di testo preferito per creare il file di configurazione:

**sudo nano /etc/httpd/conf.d/exercise4.conf**

All'interno del file, inserisci il seguente contenuto:

**<VirtualHost \*:80>**

**ServerName www.exercise4.myexam.com**

**DocumentRoot /var/www/html**

**RewriteEngine On**

**RewriteCond %{HTTP\_HOST} ^www\.exercise4\.myexam\.com$**

**RewriteRule ^/$ /exercise4.html [L]**

**</VirtualHost>**

Crea il file exercise4.html nella directory /var/www/html. Questo file conterrà il testo "Hello exercise4":

**sudo nano /var/www/html/exercise4.html inserisci il testo "Hello exercise4" nel file, salvalo e chiudilo.**

Crea il file di configurazione predefinito:

**sudo nano /etc/httpd/conf.d/default.conf**

All'interno del file, inserisci il seguente contenuto:

**<VirtualHost \*:80>**

**DocumentRoot /var/www/html**

**</VirtualHost>**

Per risolvere www.exercise4.myexam.com all'IP privato del sistema, aggiungi un'entry al file /etc/hosts:

**sudo nano /etc/hosts**

Aggiungi la seguente linea alla fine del file:

**<Indirizzo IP privato>** [**www.exercise4.myexam.com**](http://www.exercise4.myexam.com)

sostituisci <Indirizzo IP privato> con l'effettivo indirizzo IP privato del tuo sistema.

Ora che hai configurato Apache, riavvialo per applicare le modifiche

**sudo systemctl restart httpd**

## Bash script

**cd /exam/exercise5**

**nano bash\_wrapper.sh**

**#!/bin/bash**

**# Verifica che siano stati passati due parametri**

**if [ "$#" -ne 2 ]; then**

**echo "Errore: Si prega di fornire esattamente due parametri."**

**exit 1**

**fi**

**# Assegna i parametri a variabili**

**comando="$1"**

**argomento="$2"**

**# Verifica che il comando esista nel sistema**

**if ! command -v "$comando" &> /dev/null; then**

**echo "Errore: Il comando '$comando' non esiste nel sistema."**

**exit 1**

**fi**

**# Esegui il comando con l'argomento specificato**

**"$comando" "$argomento"**

**chmod +x /exam/exercise5/bash\_wrapper.sh**

## Docker

All'interno della directory /exam/exercise6, crea un file chiamato Dockerfile. Questo file definirà come costruire la tua immagine Docker personalizzata. Di seguito è riportato un esempio di Dockerfile:

**# Usa un'immagine di base a tua scelta**

**FROM ubuntu:latest**

**# Copia lo script di entrypoint e la directory loggerdir nella directory /app del container**

**COPY logger.sh /app/**

**COPY loggerdir /app/loggerdir/**

**# Imposta la variabile d'ambiente NUMBER con un valore predefinito**

**ENV NUMBER=5**

**# Definisci lo script di entrypoint**

**ENTRYPOINT ["/app/logger.sh"]**

Crea uno script Bash chiamato logger.sh nella stessa directory /exam/exercise6

**#!/bin/bash**

**# Imposta il numero di volte in cui stampare il messaggio (usa il valore predefinito se non specificato)**

**COUNT=${NUMBER:-5}**

**# Loop per stampare il messaggio nel file logger.log**

**for ((i=1; i<=$COUNT; i++)); do**

**echo "hello docker" >> /app/loggerdir/logger.log**

**done**

**echo "Done."**

**chmod +x logger.sh**

Crea un file docker-compose.yml nella directory /exam/exercise6 con il seguente contenuto:

**version: '3'**

**services:**

**myapp:**

**build: .**

**environment:**

**NUMBER: 10 # Modifica il numero a tuo piacere**

**volumes:**

**- ./loggerdir:/app/loggerdir**

Ora puoi costruire l'immagine Docker eseguendo il seguente comando nella directory /exam/exercise6, dove si trova il tuo Dockerfile:

**docker-compose build**

Una volta che l'immagine è stata costruita, puoi eseguire il container con il seguente comando:

**docker-compose up**

06/06/23

# Crea una lista di occorrenze senza ripetizioni utilizzando shell expansion

seq 1 1000 > /exam/exercise1/text\_file

# Crea un file di testo contenente i comandi

echo "wc -l /exam/exercise1/text\_file" > /exam/exercise1/command

echo "tail -n 100 /exam/exercise1/text\_file" >> /exam/exercise1/command

echo "head -n $(($(wc -l < /exam/exercise1/text\_file) / 2)) /exam/exercise1/text\_file" >> /exam/exercise1/command

# Esegui i comandi presenti nel file "command"

chmod +x /exam/exercise1/command

/exam/exercise1/command

------------------------------------------------------------------------------------------------

echo 'export HELLO\_USER="hello student"' >> /home/student/.bashrc

echo 'export HELLO\_ALL\_USERS="hello all users"' >> /etc/profile

source /etc/profile

source /home/student/.bashrc

------------------------------------------------------------------------------------------------

# Crea il gruppo 'months'

groupadd months

# Crea l'utente 'june' appartenente al gruppo 'months' e con la directory home specificata

useradd -m -d /home/months/june -g months june

# Imposta i permessi della home directory di june

chmod 700 /home/months/june

echo "UMASK 077" >> /etc/login.defs

chown june:months /home/months/june

------------------------------------------------------------------------------------------------

# Aggiorna l'elenco dei pacchetti

sudo apt update

# Installa Apache

sudo apt install apache2

sudo nano /etc/apache2/sites-available/exercise4.conf

---------

<VirtualHost \*:80>

ServerName www.exercise4.myexam.com

DocumentRoot /var/www/exercise4

<Directory /var/www/exercise4>

Require all granted

</Directory>

</VirtualHost>

---------

sudo mkdir /var/www/exercise4

cd /var/www/exercise4

echo "Hello class!" > index.html

sudo ln -s /usr/share/doc/apache2-doc/manual/ /var/www/exercise4/manual

# Abilita il sito appena creato

sudo a2ensite exercise4.conf

# Riavvia Apache per applicare le modifiche

sudo systemctl restart apache2

------------------------------------------------------------------------------------------------

nano /exam/exercise5/prime\_numbers.sh

---------

#!/bin/bash

if [ "$#" -lt 2 ]; then

echo "Errore: Devono essere passati almeno due parametri in ingresso."

exit 100

fi

for number in "$@"; do

if ! [[ "$number" =~ ^[0-9]+$ ]]; then

echo "$number non è un numero."

else

is\_prime=true

for ((i = 2; i <= number / 2; i++)); do

if [ $((number % i)) -eq 0 ]; then

is\_prime=false

break

fi

done

if [ "$number" -eq 1 ]; then

is\_prime=false

fi

if $is\_prime; then

echo "$number è un numero primo."

else

echo "$number non è un numero primo."

fi

fi

done

---------

chmod +x /exam/exercise5/prime\_numbers.sh

/exam/exercise5/prime\_numbers.sh 7 15 abc 23

------------------------------------------------------------------------------------------------

mkdir -p /exam/exercise6

# Usa un'immagine di base

FROM ubuntu:latest

# Aggiorna il sistema e installa il software necessario

RUN apt-get update && \

apt-get install -y <software\_a\_piacere>

# Copia lo script di entrypoint

COPY dockerexam.sh /dockerexam.sh

# Imposta l'entrypoint script come eseguibile

RUN chmod +x /dockerexam.sh

# Esegui lo script di entrypoint

CMD ["/dockerexam.sh"]

nano /exam/exercise6/dockerexam.sh

#!/bin/bash

# Avvia un processo a piacere (es. sleep) per mantenere il container in esecuzione

<comando\_a\_piacere> &

# L'ampersand (&) fa sì che il comando venga eseguito in background

sleep infinity &

# Attendi il segnale di terminazione

trap "exit 0" SIGTERM

# Attendi indefinitamente (puoi sostituire con ulteriori comandi o cicli)

while true; do

sleep 1

done

docker build -t exam/exercise6:1.0 /exam/exercise6

docker run -d --name exercise6\_container exam/exercise6:1.0

12/04/23

# Regular Expressions

Contengono un numero:

Espressione Regolare: [0-9]

Comando grep: **grep -E '[0-9]' /exam/exercise1/text\_file > /exam/exercise1/regex\_file**

Inizio riga con la parola "Listen":

Espressione Regolare: ^Listen

Comando grep: **grep -E '^Listen' /exam/exercise1/text\_file >> /exam/exercise1/regex\_file**

Finiscono con la parola "conf":

Espressione Regolare: conf$

Comando grep: **grep -E 'conf$' /exam/exercise1/text\_file >> /exam/exercise1/regex\_file**

Contengono il carattere 'a' oppure 'b':

Espressione Regolare: [ab]

Comando grep: **grep -E '[ab]' /exam/exercise1/text\_file >> /exam/exercise1/regex\_file**

# Alias command and environment

Aprire il file di configurazione degli alias di sistema. In Fedora, comunemente usiamo /etc/bashrc per definire alias di sistema, che saranno disponibili per tutti gli utenti.

**sudo nano /etc/bashrc**

Aggiungi l'alias p-path al file. L'alias p-path è definito come segue:

**alias p-path='echo $PATH'**

Per rendere effettive le modifiche, puoi riavviare la shell eseguendo il comando source seguito dal file /etc/bashrc:

**source /etc/bashrc**

# File system permission and user

Creare l'utente "mark" e assegnarlo al gruppo primario "exam":

**sudo useradd -m -G exam mark**

-m crea una home directory per l'utente "mark" sotto /home/mark.

-G exam assegna "exam" come gruppo primario.

Impostare le autorizzazioni predefinite in modo che i nuovi file e le directory creati dall'utente "mark" siano accessibili solo da owner e group:

**sudo chmod 770 /home/mark**

Consentire all'utente "mark" di accedere ai file e alle directory appartenenti al gruppo "students":

**sudo usermod -a -G students mark**

-a aggiunge "students" ai gruppi supplementari di "mark".

Impostare la necessità di cambiare la password al primo accesso:

**sudo chage -d 0 mark**

Impostare il cambiamento periodico della password ogni 60 giorni:

**sudo chage -M 60 mark**

Impostare il permesso SGID sulla home directory dell'utente "mark" in modo che i nuovi file creati appartengano al gruppo "students":

**sudo chmod g+s /home/mark**

# HTTPD

Installa Apache HTTP Server (HTTPD) e PHP

**sudo dnf install httpd php**

Abilita il servizio HTTPD in modo che venga avviato automaticamente all'avvio del sistema:

**sudo systemctl enable httpd**

Modifica il file di configurazione di Apache per farlo ascoltare sulla porta 8090:

**sudo nano /etc/httpd/conf/httpd.conf**

Trova la direttiva Listen 80 e cambiala in Listen 8090

Crea la regola del firewall per consentire l'accesso esterno alla porta 8090:

**sudo firewall-cmd --zone=public --add-port=8090/tcp --permanent**

**sudo firewall-cmd –reload**

Imposta la DocumentRoot per il tuo servizio su /exam/exercise4. Modifica il file di configurazione del sito virtuale di default:

**sudo nano /etc/httpd/conf.d/welcome.conf**

All'interno di questo file, trova la direttiva DocumentRoot e cambiala in:

**DocumentRoot "/exam/exercise4"**

Crea il file /exam/exercise4/index.php con il contenuto desiderato:

**sudo mkdir -p /exam/exercise4**

**sudo nano /exam/exercise4/index.php**

All'interno di index.php, inserisci il seguente codice PHP:

**<?php**

**echo "Hello Exam!!"; (te lo da l’esercizio)**

**?>**

Abilita il modulo PHP per Apache:

**sudo ln -s /etc/httpd/conf.modules.d/15-php.conf /etc/httpd/conf.d/15-php.conf**

Avvia il servizio HTTPD:

**sudo systemctl start httpd**

# Bash script

**mkdir -p /exam/exercise5**

**cd /exam/exercise5**

**nano calc.sh**

Salva questo script in un file chiamato calc.sh sotto la directory /exam/exercise5 e rendilo eseguibile:

**chmod +x /exam/exercise5/calc.sh**

docker-composeCrea una directory di lavoro per il tuo progetto Docker, ad esempio my\_web\_app.

**mkdir my\_web\_app**

**cd my\_web\_app**

All'interno della directory my\_web\_app, crea una directory chiamata content. Questa sarà la directory in cui verranno montati i contenuti web.

**mdkir -p /exam/exercise6/content**

Crea un file chiamato Dockerfile con il seguente contenuto:

**nano Dockerfile**

**# Usa l'immagine base Fedora**

**FROM fedora:latest**

**# Installa Apache e altri pacchetti necessari**

**RUN dnf install -y httpd**

**# Copia il contenuto web nella DocumentRoot di Apache**

**COPY ./content /var/www/html**

**# Esponi la porta 80**

**EXPOSE 80**

**# Avvia Apache in background**

**CMD ["/usr/sbin/httpd", "-D", "FOREGROUND"]**

Ora, crea un file docker-compose.yml

**version: '3'**

**services:**

**my\_web\_app:**

**build: .**

**ports:**

**- "8000:80"**

**volumes:**

**- ./content:/var/www/html**

Crea uno script chiamato entrypoint.sh con il seguente contenuto:

**#!/bin/bash**

**if [ ! -f /var/www/html/index.html ]; then**

**echo "Creating index.html with start date..."**

**echo "Start Date: $(date)" > /var/www/html/index.html**

**fi**

**# Esegui l'HTTPD**

**exec "$@"**

Assicurati che lo script entrypoint sia eseguibile:

**chmod +x entrypoint.sh**

Ora puoi avviare il tuo servizio web con Docker Compose:

**docker-compose up -d**