

Connessione alla propria macchina

1. Aprire il client SSH .
2. Posizionarsi nella directory di salvataggio del file .pem allegato alla mail.
3. Tramite il client SSH, utilizza il seguente comando al fine di impostare le autorizzazioni del file della chiave privata in **read only** per l'utente owner.
chmod 400 keyPairName.pem
4. Nella finestra del terminale, utilizzare il comando ssh per connettersi all'istanza. Specificare il percorso e il nome del file della chiave privata (.pem), il nome utente per l'istanza e il nome DNS pubblico per l'istanza.
5. Utilizzare la username **centos** per la connessione al sistema

Esempio:

```
ssh -i "keyPairName.pem"
```

```
centos@ec2-xx-xxx-xx-x.eu-central-1.compute.amazonaws.com
```

Per passare all'utente **root** utilizzare il comando "**sudo su -**"

```
[centos@ip-xxx-xx-xx-xxx ~]$ sudo su -
```

Exercise 1: Managing pipeline and regular expression

- Utilizzare il comando **find**, e l'utente **student**, per trovare tutti i files e directories sotto il path **/usr** che contengono almeno un numero nel nome. Redirigere lo standard output sul file **/exam/exercise1/find.txt**.

Exercise 2: Special file permission

- Creare una directory sotto **/exam/exercise2** dove gli utenti che possono accedere al gruppo **collaboration** potranno condividere files
- Tutti i file creati sotto la directory **/exam/exercise2** dovranno essere assegnati automaticamente al gruppo **collaboration**

Exercise 3: Default file permission

- Creare l'utente **exercise3** appartenente al gruppo **exams**.
- Fare in modo che nuovi file e directory creati dall'utente **exercise3** alla creazione, di default non abbiano nessun permesso associato agli utenti **others**

Exercise 4: Shell environment and alias command

- Creare un nuovo comando o alias command chiamato **userinfo** disponibile al login per **TUTTI** gli utenti del sistema. Chiamato dovrà stampare la seguente stringa
user: <username> - working directory: <print working directory> - home_directory: /home/directory

Esempio:

```
[student@hostname tmp]$ userinfo
user: student - working directory: /tmp - home_directory:
/home/student
```

Exercise 5: Systemd

- Installare sul vostro sistema il servizio **vsftpd**
- Il servizio deve essere attivo al boot della macchina

Exercise 6: Bash script

- Create uno script bash sotto **/exam/exercise6** chiamato **users.sh** con le seguenti caratteristiche:
 - accetti in ingresso un file contenente un elenco di utenti da creare o rimuovere
 - gli utenti nel file saranno presenti in colonna
esempio:
mrossi
abianchi
lverdi
jsmith
 - l'azione da intraprendere (**add/remove**)
 - se **add** l'utente verrà aggiunto al sistema
 - se **remove** l'utente verrà rimosso dal sistema
 - Lo script dovrà prevedere una serie di controlli:
 - se l'utente da inserire è già presente non dovrà tentare la creazione dell'utente
 - se l'utente da rimuovere non è presente non dovrà tentare la rimozione dell'utente
 - se i parametri passati in ingresso non sono quelli attesi restituisca il messaggio
Usage: **users.sh userfile <add|remove>**
 - Se il file passato in ingresso non esiste restituisca l'errore
Error: file <userfileName> not found

Exercise 7: Docker

- La directory **/exam/exercise7** dovrà contenere i files:
 - **Dockerfile**
 - **template.html**
 - **entrypoint.sh**
 - **docker-compose.yml**
- **template.html** conterrà la index del server HTTPD con una variabile di ambiente modificabile in fase di start container
il suo contenuto sarà:

```
<html>
  <head>
    <title>Exam</title>
  </head>
  <body>
    <b>Hello $EXAM</b>
  </body>
</html>
```
- **Dockerfile** conterrà le istruzioni per la creazione della immagine Docker basata su Centos:8 con installato il server HTTPD
- **entrypoint.sh** sarà lo script chiamato come ENTRYPOINT dalla immagine Docker, il cui compito è quello di andare a popolare la index.html con la variabile di ambiente EXAM, ed effettuare lo start del server HTTPD
- **docker-compose.yml** verrà utilizzato per effettuare la build automatica della immagine (se non presente), per lanciare l'immagine Docker, per passare la variabile di ambiente \$EXAM, per attivare port binding tra il vostro sistema di esame e la porta del servizio HTTP presente sul container.

Question :

- Si descriva a parole la differenza tra metodo dichiarativo e imperativo e quali secondo voi possono essere le principali differenze tra i due approcci in termini di ripetibilità e riutilizzo nel tempo. Salvare la risposta sotto **exam/question/question.txt**
- Si riporti inoltre un esempio di oggetto di più altro livello tra quelli visti descritto tramite file yaml (**exam.yaml** da creare nel path **/exam/question/exam.yaml**), supponendo di dover gestire una applicazione basata sulla immagine **"exam/applicationexam:latest"** con lable **exam1** su 4 repliche attive.