

## 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 lo 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: Regular expression

- dato un file con il seguente contenuto:  
can  
man  
fan  
dan  
ran  
pan
- riportare una regular expression che rispetti il seguente schema  
**Match** can  
**Match** man  
**Match** fan  
**Skip** dan  
**Skip** ran  
**Skip** pan
- potete testare il tutto creando il file `regexpr.txt` sotto `/exam/exercise1/regexpr.txt` e il comando **grep**
- riportare la soluzione sotto `/exam/exercise1/solution`

### example

```
[root@linux /]# grep RegExFound /exam/exercise1/regexpr.txt
can
man
fan
[root@linux /]# echo RegExFound > /exam/exercise1/solution
```

## 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: User and Group

- Create un nuovo gruppo **exams** con GID 4000
- Creare lo studente **exercise3** appartenente al gruppo **exams** con le seguenti caratteristiche:
  - UID 4100
  - home directory **/home/exams/exercise3**
  - dovrà cambiare password al primo accesso
  - l'account scadrà ad un anno dalla sua creazione
  - dovrà poter accedere a file e directory appartenenti al gruppo **users**
  - Fare in modo che nuovi file e directory creati dall'utente **exercise3**, di default (al login quindi), possano essere letti scritti o visti, da tutti gli utenti del sistema (presenti e futuri)

### 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 4: HTTPD - SYSTEMD

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che HTTPD venga lanciato al boot della macchina
- Il servizio sarà in ascolto sulla **SOLA** porta **8080**

## Exercise 6: Bash script

- Create uno script **bash** sotto **/exam/exercise6** chiamato **checkusername.sh** con le seguenti caratteristiche:
  - accetti in ingresso una lista a piacere di username da controllare
    - minimo 1 username
    - massimo 5 username
  - Stampi a video per tutti gli username passati nella lista le seguenti informazioni:
    - Se lo username non è presente un messaggio: "user <username> not found"
      - nessun messaggio di errore (STDERR) di sistema deve comparire a video.
    - Se lo username è presente il messaggio: "user <username> found:"
      - l'output del comando **id <username>**
      - l'output del comando **chage -l <username>**
      - la riga contenente le sole informazioni username presenti sotto **/etc/passwd**
  - Lo script dovrà prevedere una serie di controlli:
    - se non viene passato nessun parametro lo script esca con **exit code 10** e riporti il messaggio di errore: **no parameters passed - Usage: checkusername.sh <username list>**
    - Se il numero di parametri eccede il numero massimo lo script esca con **exit code 20** un messaggio di errore: **too many parameters passed (max 5) - Usage: checkusername.sh <username list>**
    - **NB. Il messaggio di errore DEVE essere gestito tramite una unica funzione**

Esempio:

```
[root@linux /]# bash checkusername.sh
no parameters passed
Usage: checkusername.sh <username list>
[root@linux /]# echo $?
10
----

[root@linux /]# bash checkusername.sh luca mario elena lucia emma
teo
too many parameters passed (max 5)
Usage: checkusername.sh <username list>
[root@linux /]# echo $?
20
----

bash checkusername.sh elena luca
user elena not found
user luca found:
uid=1000(luca) gid=1000(luca) groups=1000(luca)
Last password change                : Aug 21, 2022
Password expires                     : never
```

```
Password inactive : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
luca:x:1000:1000::/home/luca:/bin/bash
```

## Exercise 7: Docker

- La directory **/exam/exercise7** dovrà contenere i files:
  - **Dockerfile**
  - **hello.sh**
  - **docker-compose.yml**
- **Dockerfile** conterrà le istruzioni per la creazione della immagine Docker basata sull'immagine di base **cavatortaluca/exam:centos8**
- **hello.sh** è uno script bash il cui compito sarà quello allo start del container di stampare il messaggio "hello EXAM" e uscire.
  - La parola EXAM potrà essere modificata (quindi gestita) tramite variabile di ambiente passata allo start del container
- **docker-compose.yml** verrà utilizzato per:
  - effettuare la build automatica dell'immagine (se non presente)
  - lanciare l'immagine Docker
  - passare la variabile di ambiente EXAM

## Kubernetes :

- Cosa si intende per **Observability**.
- Quale funzionalità/comportamento è **a vostro parere** il più importante, associato al concetto di observability e a un cluster Kubernetes?
- Salvare la risposta sotto **/exam/question/question.txt**