

## 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 Files with Shell Expansion and Command substitution

- Utilizzare l'account **student** (se non presente crearlo)
- Creare sotto il path **/exam/exercise1/** le directory **directoryX** con **X** compreso tra 1 e 20
- all'interno di ogni directory sotto **/exam/exercise1/directoryX** creare i files
  - **file(a..z)\_DATE.txt**
    - con **(a..z)**= tutte le lettere dalla a alla z
    - **DATE** la data di creazione file nel formato Y-m-d (**date +%Y-%m-%d**)

Il risultato sarà il seguente:

```
|-- exercisel_directory1
|   |-- filea_2021-04-07
|   |-- fileb_2021-04-07
|   `-- filec_2021-04-07
etc...
|-- exercisel_directory2
|   |-- filea_2021-04-07
|   |-- fileb_2021-04-07
|   `-- filec_2021-04-07
etc...
```

## Exercise 2: Managing pipeline and regular expression

- Utilizzare il comando **ls** o **find**, e l'utente **student**, per trovare tutti i file e directory sotto la directory **/var** che contengono il pattern **cat**. Redirigere lo standard output sul file **/exam/exercise2/cat.txt**.
- Fare in modo che eventuali errori non compaiano nel file cat.txt

## Exercise 3: alias command

- Creare i seguenti aliases commands disponibili **a tutti** gli utenti del sistema:
  - **"c"**
    - comando: **clear**
  - **"hwmp"**
    - comando: **uname -mp**
- Creare un alias command disponibile a tutti gli utenti del sistema chiamato **"home"** il cui compito sia restituire la home directory dell'utente che lancerà il comando.

## Exercise 4: User and Group

- Creare due nuovi gruppi chiamati Marvel e Disney
  - marvel con GID 3000
  - disney con GID 4000
- Creare gli utenti appartenenti al gruppo Marvel **thor**, **wolverine** e **hulk**
  - l'utente **thor** avrà le seguenti caratteristiche:
    - UID 3001
    - home directory /home/marvel/thor
    - password thor
    - dovrà cambiare password una volta ogni due mesi
  - l'utente **wolverine** avrà le seguenti caratteristiche:
    - UID 3002
    - potrà accedere e gestire i file e directory appartenenti al gruppo disney
    - password wolverine
    - dovrà cambiare password al primo accesso e successivamente una volta ogni 3 mesi
    - l'account scadrà dopo un anno dalla sua creazione
  - l'utente **hulk** avrà le seguenti caratteristiche:
    - UID 3003
    - home directory /home/marvel/hulk
    - password hulk
    - l'utente **non** potrà fare accesso al sistema
- Creare gli utenti **bb-8** e **yoda** appartenenti al gruppo **disney** entrambi con password disney, gruppo primario **disney** e gruppo secondario **marvel** e **users**.

## Exercise 5: File permission

- Creare l'utente **linux** appartenente al gruppo **opensource**.
- Fare in modo che nuovi files e directory creati dall'utente **linux** alla creazione, di **default** non possano essere letti scritti o visti, da nessuno al di fuori dell'utente **linux**.

## Exercise 6: Bash script

- Create uno script chiamato **/exam/exercise6/exercise6.sh**
  - Lo script accetterà come unico argomento un file
  - Se il file passato non è un file o non esiste restituirà in output **"File not found"**
  - Se il file passato esiste restituisca il formato del file
  - Se non viene passato nessun argomento dovrà restituire il messaggio **"Usage:/exam/exercise6/exercise6.sh <file>"**

Esempio:

```
bash /exam/exercise6/exercise6.sh fileName
fileName: ASCII text, with escape sequences
```

## Exercise 7: Bash script

- Create uno script chiamato **/exam/exercise7/users.sh**
  - Lo script accetterà come primo argomento quale azione lo script dovrà eseguire (**list/add/remove**) e come secondo un nome utente
  - Se chiamato con l'opzione **list** dovrà restituire la lista degli utenti presenti sul sistema
  - Se chiamato con l'opzione **add** dovrà aggiungere l'utente passato come argomento
  - Se chiamato con l'opzione **remove**, dovrà eliminare l'utente passato come argomento
  - Se lo script viene chiamato senza argomenti o errati dovrà tornare il messaggio **"Usage: /exam/exercise7/createuser.sh list/add/remove <username> "**
  - Si utilizzi il **case** statement per gestire lo script

example:

```
bash /exam/exercise7/user.sh list
root
daemon
bin
sys
sync
[..]
bash /exam/exercise7/user.sh add testuser
testuser added
bash /exam/exercise7/user.sh remove testuser
testuser removed
bash /exam/exercise7/user.sh
"Usage: /exam/exercise7/user.sh list
/add/remove <username>"
```

## Exercise 8: Firewall: Configure port forwarding

- Create una regola di port forwarding sul vostro sistema in ingresso sulla porta 8081/tcp verso la porta 22/tcp.
- Potete testare la nuova regola connettendovi via ssh al vostro sistema sulla porta 8081 con l'opzione -p 8081 del comando ssh. Attenzione ad abilitare le regole firewall sulla porta 8081 TCP per poter accedere al sistema dall'esterno.

## Exercise 9: Systemd and Firewall

- Installare sul vostro sistema il servizio **vsftpd** tramite **yum**
  - Il servizio deve essere attivato al boot della macchina
  - Individuare il **pid** del servizio tramite **systemctl**
  - verificare su quale porta il servizio è in listening con il comando

```
netstat -tulpen | grep <pid>
```

esempio:

```
netstat -tulpen | grep 12345
```

```
tcp        0      0 0.0.0.0:<PortNumber>
0.0.0.0:*          LISTEN      12345
```
- Aggiungere la porta appena individuata alle regole firewall per il protocollo **TCP**  
(ricordarsi di rendere permanente la regola firewall)

## Exercise 10: Docker

docker e docker-compose sono già installati sul vostro sistema, e il demone docker è già abilitato al boot del sistema

- Creare una nuova immagine Docker basata su centos:8 che chiamerete **exam/exercise10:1.0**
- Il compito di questa immagine una volta lanciata, sarà quello di stampare a video la stringa “Hello Exam” ogni 2 secondi, per 5 volte, e poi uscire stampando la stringa “goodbye!!”
- Fare in modo che la **frase** “Hello Exam” possa essere modificata tramite variabile di ambiente.
- Preparare sotto **/exam/exercise10** il file **docker-compose.yml** per gestire lo start della vostra immagine tramite docker-compose tool, comprensivo della gestione della variabile di ambiente.

## Question:

Si descriva a parole come in kubernetes viene gestito il workload applicativo delle applicazioni in container attraverso l'oggetto Pod. Di cosa si tratta, quali le caratteristiche principali. In che modo, e come, questo oggetto viene gestito dagli oggetti di workload di più alto livello analizzati ReplicaSet/ReplicationController.