

## 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.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 **exercise\_directoryX** con **X** = tutte le lettere comprese tra **A** e **Z**
- all'interno di ogni directory sotto **/exam/exercise1/exercise\_directoryX** creare i files
  - **fileY\_DATAMS.txt**
    - **Y** tutti i numero tra 1 e 100
    - **DATAMS** i minuti e secondi della data di creazione file nel formato M:S (**date +%M:%S**)

Il risultato sarà il seguente:

```
|-- exercise_directorya
|   |-- file1_36:47
|   |-- file2_36:47
|   `-- file3_36:47
etc...
|-- exercise_directoryb
|   |-- file1_36:47
|   |-- file2_36:47
|   `-- file3_36:47
etc...
```

## Exercise 2: Managing pipeline and regular expression

- Utilizzare l'utente **student**, per trovare tutti i file e directory sotto la directory **/var** che contengono il pattern **aXt**. con **a**=il carattere "a", **b**=il carattere "b", **X**=qualunque carattere alfabetico minuscolo tra a e z.
- Redirigere lo standard output sul file **/exam/exercise2/aXt.txt**.
- Fare in modo che eventuali errori non compaiano nel file aXt.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 dal solo utente **linux** alla creazione, di **default** non possano essere letti scritti o visti, da nessuno al di fuori dell'utente **linux**.

example:

```
[linux@desktop ~]$ ls -l
total 2
drwx----- 2 linux users 4096 Sep  9 14:30 directory
-rw----- 1 linux users    0 Sep  9 14:30 file
```

## Exercise 6: HTTPD Dynamic content

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che HTTPD venga lanciato al boot della macchina
- Fare in modo che il servizio HTTPD risponda **SOLAMENTE** sulla porta **8080**
- Aggiungere la regole firewall per poter accedere dall'esterno al servizio HTTPD porta **8080**
- Il web server dovrà erogare contenuti dinamici utilizzando il linguaggio di scripting PHP
- Fare in modo che la Document Root impostata per il vostro servizio sia `/exam/exercise6/`
- Creare il file `/exam/exercise6/index.php` con il seguente contenuto

```
<?php
$httpd_user = shell_exec('whoami');
echo "<H1>Hello from $httpd_user user</H1>";
?>
```
- Potete verificare che il tutto funzioni collegandovi all'indirizzo IP della vostra macchina AWS dal browser locale alla vostra postazione:
  - `http://ec2-xx-xxx-xx-x.eu-central-1.compute.amazonaws.com:8080/index.php`

## Exercise 7: Bash script

- Creare uno script bash chiamato `/exam/exercise7/fact.sh` con il seguente comportamento:
  - Accetti in ingresso un numero e ne calcoli il fattoriale

ex:

```
bash /exam/exercise7/fact.sh 5
120
```

## Exercise 8: Bash script

- Create uno script chiamato **/exam/exercise8/manageusers.sh**
  - Lo script accetterà come primo argomento un nome utente e come secondo quale azione lo script dovrà eseguire (**add/remove/info**)
  - 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 passato con l'opzione **info**, dovrà restituire le informazioni utente (UID GID GROUPS)
  - Se lo script viene chiamato senza argomenti dovrà tornare il messaggio

**"Usage: /exam/exercise8/manageuser.sh <username> add/remove/info"**

example:

```
bash /exam/exercise8/createuser.sh testuser add
bash /exam/exercise8/createuser.sh testuser info
uid=1001(testuser) gid=1001(testuser) groups=1001(testuser)
bash /exam/exercise8/createuser.sh testuser remove
bash /exam/exercise8/createuser.sh testuser info
id: 'testuser': no such user
bash /exam/exercise7/createuser.sh testuser
"Usage: /exam/exercise7/createuser.sh <username> add/remove"
```

## Exercise 9: Firewallld

- Create una nuova regola firewall per il servizio http (in più a quella inserita nell'esercizio 6) che vada ad aggiungere ai log delle connessioni, verso il servizio http il prefisso **"Exercise9"**.
- Limitate il log ad un massimo di 5 nuovi messaggi per secondo

## Exercise 10: Docker

- Creare la propria immagine Docker basata su **centos:8**
- La directory **/exam/exercise10** dovrà contenere i files:
  - **Dockerfile**
  - **hello.sh**
- **hello.sh** sarà lo script bash il cui compito sarà quelli di stampare “hello <name>!!” il parametro name dovrà essere una variabile di ambiente contenente di default il valore “exam”
- L'immagine dovrà chiamarsi `exam/myhelloexam:1.0`

Risultato:

```
docker run exam/myhelloexam:1.0
```

```
Hello Exam!!
```

```
docker run -e ENV_VAR_NAME="Luca" exam/myhelloexam:1.0
```

```
Hello Luca!!
```

### Question

- In una infrastruttura (virtuale o fisica che sia) basata su tecnologia a container gestiti all'interno di un orchestratore quale è secondo voi il principale beneficio dal punto di vista funzionale, gestionale.
- Salvare la risposta in un file sotto **/exam/question/**