

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

- Creare la seguente struttura sotto le directory **/exam/exercise1/exercise1_dirX/fileY** con **X** compreso tra 1 e 10 e **Y** compreso tra 10 e 30

```
exercise1_dir1
|
|----- file10.txt
|----- etc..
|----- file20.txt
exercise1_dir2
|
|----- file1.txt
etc...
```

Exercise 2: Managing regular expression and command substitution

- Utilizzare l'utente **student** per trovare tutti, e solamente i file (non le directory) presenti su tutto il file system (/) che contengono il pattern "**text**" (si consiglia di utilizzare il comando **find**). Per ogni file trovato determinare il **file type** utilizzando il comando **file** e reindirizzare l'output su **/exam/exercise2/find.txt**. Eventuali errori andranno rediretti sul file **/exam/exercise2/error.txt**

Esempio di contenuto del file find.txt:

/sys/module/xfrm_user/sections/.text:	regular file, no read permission
/usr/share/groff/1.22.3/font/devps/generate/textmap:	C source, ASCII text
/usr/share/groff/1.22.3/font/devps/text.enc:	ASCII text

Exercise 3: User Group and default permission

- Create l'utente **thomas** con le seguenti caratteristiche:
 - **UID 2000**
 - home directory **/home/exam/thomas**
 - dovrà cambiare password una volta al mese
 - dovrà cambiare password al primo login
 - dovrà poter accedere a file e directory appartenenti al gruppo **students**
 - l'account scadrà dopo due anni dalla sua creazione
 - fare in modo che tutti i file e directory creati dall'utente **thomas** di default non abbiano nessun permesso (rwx) per gli utenti **other**

Exercise 4: File permission

- Creare la directory **/exam/exercise4/collaboration/students** dove gli utenti facenti parte del gruppo **students** potranno condividere file e directory
- Fare in modo che tutti i file e directory create sotto **/exam/exercise4/collaboration/students** vengano assegnati automaticamente al gruppo **students**

esempio:

```
[root@desktop ~]# cd /exam/exercise4/collaboration/students
[root@desktop students]# touch example-root
[root@desktop students]# ls -la example-root
-rw-r--r-- 1 root students 0 Feb 10 22:09 example-root
[root@desktop students]# su - thomas
[thomas@desktop ~]$ cd /exam/exercise4/collaboration/students
[thomas@desktop thomas]$ touch example-thomas
[thomas@desktop thomas]$ ls -la example-thomas
-rw-r----- 1 thomas students 0 Feb 10 22:10 example-thomas
```

Exercise 5: Environment

- Creare un nuovo script chiamato **hello** per l'utente **student** il cui compito sarà quello di stampare in output la stringa "**hello student**"
- Posizionare il vostro script sotto la directory **/exam/exercise5/bin**
- Modificare l'environment dell'utente student in modo da poter richiamare il comando senza specificare il path **/exam/exercise5/bin/hello** (la modifica dell'environment dovrà essere resa persistente al login dell'utente utente).
- **Non si utilizzi un alias command**

esempio:

```
$ su - student
$ hello
hello student
$ cd /etc
$ hello
hello student
```

Exercise 6: Firewall: Configure port forwarding

- Create una regola di port forwarding sul vostro sistema in ingresso sulla porta 8080/tcp verso la porta 22/tcp.
- Potete testare la nuova regola facendo connessione via ssh al vostro sistema sulla porta 8080 con l'opzione -p 8080 del comando ssh.

Exercise 7: Bash script

- Create uno script bash chiamato **/exam/exercise7/print.sh** con il seguente comportamento:
 - accetti in ingresso N parametri
 - stampi in output tutti i parametri passati in ingresso come nell'esempio riportato:

example:

```
[root@desktop ~]# bash print.sh hello luca 1 2 3
hello luca 1 2 3
[root@desktop ~]# bash print.sh "hello luca" 1 2 3
hello luca 1 2 3
```

Exercise 8: Bash script

- Creare uno script chiamato **/exam/exercise8/createuser.sh**
- Lo script accetti in ingresso due argomenti
 - 1 - un file contenente un elenco di utenti in colonna nel formato
nome:cognome:uid
nome:cognome:uid
 - 2 - l'azione da intraprendere:
add/remove
- Lo script dovrà creare o rimuovere gli utenti presenti all'interno del file di testo nel seguente formato:
 - **username** composta dalla iniziale del nome e il cognome per intero
(mario:rossi:3000 username=**mrossi**)
 - **UID** lo uid passato (mario:rossi:**3000** uid=**3000**)
 - Utilizzare UID >= 4000
- Predisporre alcuni check che riportino i seguenti errori:
 - argomenti passati diversi da due, esca con il messaggio "**usage: createuser.sh <filename> <add/remove>**"
 - file passato non trovato: "**error: <filename> not found**"
- Redirigere lo standard error dei comandi utilizzati per la creazione o rimozione utenti su **/dev/null**
- Procedere alla creazione o alla eliminazione dell'utente solo nel caso in cui l'utente da creare **non** sia già presente, e se da eliminare **sia** presente sul sistema.
- Per ogni utente creato o eliminato si stampi a video il seguente messaggio:
 - Se l'utente è stato creato con successo: **Utente: NomeUtente creato con successo**
 - Se l'utente è stato rimosso con successo: **Utente: NomeUtente rimosso con successo**

Exercise 9: HTTPD

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che il servizio sia attivo al boot della macchina
- Fare in modo che il servizio Apache sia in ascolto solo sulla porta **8081** e aggiungete la regola firewall corretta per poter accedere dall'esterno al servizio.
- Fare in modo che la Document Root impostata per il vostro servizio sia **/exam/exercise9/**
- Creare il file **/exam/exercise9/index.html** con il seguente contenuto

```
<html>
  <head>
    <title>Exam 5 Exercise9</title>
  </head>
  <body>
    Hello Exam
  </body>
</html>
```
- Potete verificare che il tutto funzioni dal browser locale alla vostra postazione collegandovi all'indirizzo IP della vostra macchina AWS:
 - <http://ec2-xx-xxx-xx-x.eu-south-1.compute.amazonaws.com:8081>

Exercise 10: Docker

- Creare il file **/exam/exercise10/docker-compose-exam.yaml** per gestire lo start del vostro container
- L'immagine dovrà attivare un server web non importa quale sia.
- Tramite compose abilitare il bind locale su porta **80** e porta in ascolto sul container.
- Il servizio in container dovrà essere fruibile dall'esterno e restituire il contenuto statico:

```
Hello Docker!!
```
- Potete verificare che il tutto funzioni dal browser locale alla vostra postazione collegandovi all'indirizzo IP della vostra macchina AWS:
 - <http://ec2-xx-xxx-xx-x.eu-south-1.compute.amazonaws.com:80>
- Attenzione alle regole firewall per l'accesso dall'esterno.

Question

- Si descriva a parole la differenza tra metodo dichiarativo e imperativo riportando un esempio associato alla creazione di oggetti all'interno del cluster Kubernetes in modalità dichiarativa e imperativa
- Salvare la risposta sotto **/exam/question/**