

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 -
```

N.B: LA DIRECTORY DI PARTENZA DA CUI SVOLGERE GLI ESERCIZI QUANDO RICHiesto È **/exam**

- NON **/root/exam**
- NON **/home/student/exam**
- NON QUALUNQUE COSA NON SIA **/exam/exercise...**

Exercise 1: Managing Files with Shell Expansion

- Creare la seguente struttura sotto le directory **/exam/exercise1/exercise1_dirX/file-Y-DATE** con:
 - **X** compreso tra 1 e 10
 - **Y** compreso tra 1 e 100
 - **DATE** la data di creazione nel formato **Y-m-d**

```
exercise1_dir1
|
|----- file-1-2022-09-02
|----- etc..
|----- file-100-2022-09-02
exercise1_dir2
|
|----- file-1-2022-09-02
etc...
```

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 i nuovi gruppo **exams** con GID 4000 e **students** con GID 4001
- Creare lo studente **exercise** con le seguenti caratteristiche:
 - UID **4100**
 - gruppo primario **exams**
 - gruppo secondario **students**
 - dovrà cambiare password al primo accesso
 - l'account scadrà a 6 mesi dalla sua creazione
 - Fare in modo che nuovi file e directory creati dall'utente **exercise**, 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 alias command chiamato **myuserinfo** disponibile al login per il solo utente **userinfo** (se non presente crearlo). Chiamato dovrà stampare la seguente stringa:

```
I'm userinfo, my working directory is: <print working directory> -  
my environment variables are: <print of userinfo environment  
variables>
```

Exercise 5: 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**
- Document root: **/exam/exercise5/**
- Inserire le regole firewall per poter accedere al servizio sulla porta 8080 dall'esterno
 - potete testate il tutto collegandovi via browser all'indirizzo pubblico, lo stesso utilizzato per la connessione ssh:
http://ec2-xx-xxx-xx-x.eu-central-1.compute.amazonaws.com:8080

Exercise 6: Bash script

- Create uno script **bash** sotto **/exam/exercise6** chiamato **exercise5index.sh** con le seguenti caratteristiche:
 - accetti in ingresso
 - una stringa di testo
 - un comando per definire il comportamento dello script
 - Stringa di testo:
 - lunghezza a piacere su un'unica riga
 - Comando
 - append:
 - se **append**: lo script andrà ad aggiungere la stringa di testo al file **/exam/exercise5/index.html**
 - new
 - se **new**: lo script andrà a sovrascrivere tutto il contenuto del file **/exam/exercise5/index.html** con la stringa di testo passata in ingresso
 - Lo script dovrà prevedere il controllo:
 - passato nessuno o un solo parametro lo script esca con **exit code 1** e riporti il messaggio: **Usage: exercise5index.sh "<string>" <command>**
 - Il messaggio di errore DEVE essere gestito tramite **funzione**

Esempio:

```
[root@linux /]# bash exercise5index.sh "hello world"
Usage: exercise5index.sh "<string>" <command>
[root@linux /]# echo $?
1
----
[root@linux /]# bash exercise5index.sh "hello world" new
[root@linux /]# cat /exam/exercise5/index.html
hello world
[root@linux /]# bash exercise5index.sh "hello exam" append
[root@linux /]# cat /exam/exercise5/index.html
hello world
hello exam
[root@linux /]# bash exercise5index.sh "hello students!!!!" new
[root@linux /]# cat /exam/exercise5/index.html
hello students!!!!
```

Exercise 7: Docker

- La directory **/exam/exercise7** dovrà contenere i files:
 - **Dockerfile**
 - **docker-compose.yml**
- **Dockerfile** conterrà le istruzioni per la creazione della immagine Docker:
 - L'immagine dovrà erogare le funzionalità di un web server a piacere (Apache/Nginx ecc...)
 - Potete utilizzare qualunque immagine, anche immagini con web server già installato e configurato.
 - Se volete utilizzare centos:8 come immagine di partenza su cui installare il servizio utilizzate **cavatortaluca/exam:centos8** dove ho già predisposto la fix per EOL di centos:8
- **docker-compose.yml** verrà utilizzato per:
 - effettuare la build dell'immagine
 - lanciare l'immagine Docker
 - definire le una porta locale al sistema in bind con la porta in ascolto del servizio web in containe
 - passare la DocumentRoot al servizio in container con un bind locale a partire dalla directory **/exam/exercise5/** sulla document root del servizio in container
 - N.B. se non avere svolto l'esercizio 5 o 6 create una index.html a piacere sotto **/exam/exercise5/index.html**

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**