Connessione alla propria macchina

To access your instance:

1. Open an SSH client.
2. Locate your private key file (yourname-exam7.pem).
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:
   chmod 400 name-exam7.pem
4. Connect to your instance using its Public DNS:
   ec2-xx-xxx-xx-x.eu-central-1.compute.amazonaws.com
5. Use **centos** user to connect

Example:

```
ssh -i "Document/name-exam7.pem"
centos@ec2-xx-xxx-xx-x.eu-central-1.compute.amazonaws.com
```

Switch to root user

[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** compreso tra 1 e 100
- all'interno di ogni directory sotto **/exam/exercise1/exercise_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-H:M:S (**man date**)

Il risultato sarà il seguente:

```
|-- exercise1_directory1
|   |-- file a_20-09-18-00:36:47
|   |-- fileb_20-09-18-00:36:47
|   `-- filec_20-09-18-00:36:47
etc...
|-- exercise1_directory2
|   |-- filea_20-09-18-00:36:47
|   |-- fileb_20-09-18-00:36:47
|   `-- filec_20-09-18-00:36:47
etc...
```

Exercise 2: Managing pipeline and regular expression

- Trovare trovare sotto la directory **/usr** tutti i file presenti che terminano in "**.conf**". Redirigere lo standard output sul file **/exam/exercise2/ls.txt**.

Exercise 3: User and Group

- Create due nuovi gruppi **teachers** e **students**
    - **teachers** con GID 7000
    - **students** con GID 7001
- Creare l'utente appartenente al gruppo **teachers**: **cavatorta**
    - l'utente **cavatorta** avrà le seguenti caratteristiche:
        - **UID** 7000
        - password **cavatorta**
        - dovrà cambiare password una volta ogni 2 mesi e al primo accesso
        - dovrà poter accedere a file e directory di tutti gli utenti appartenenti al gruppo **students**
        - dovrà avere tra i gruppi secondari **anche** il gruppo **wheel**
        - non creare il gruppo **cavatorta**
- Creare l'utente appartenente al gruppo **students**: **patelli**
    - L'utente **patelli** avrà le seguenti caratteristiche:
        - **UID** 7010
        - l'account scadrà dopo due anni dalla sua creazione
        - dovrà avere come gruppo secondario: **users**
        - dovrà utilizzare **/bin/sh** come login shell, al posto di **/bin/bash**
        - questo utente dovrà risultare in stato di Lock (non potrà fare accesso al sistema fino a che l'utente root non disabiliterà il blocco)

Exercise 4: controlling access to file with linux file system permission

- Creare una directory chiamata **stooges** sotto il path **/exam/exercise4** all'interno della quale gli utenti **curly**, **larry** e **moe** facenti parte del gruppo **stooges** potranno lavorare in maniera collaborativa sui file creati
- Solo gli utenti e gruppo potranno accedere creare ed eliminare files contenuti in **/exam/exercise4/stooges**
- Nuovi files creati in questa directory dovranno essere assegnati automaticamente al gruppo **stooges**.
- I nuovi file creati dagli utenti **curly**, **larry** e **moe** non saranno accessibili (rwx) al di fuori del gruppo

Exercise 5: Systemd

- Visualizzare la lista di tutte le unit di tipo **socket** active e inactive presenti sul sistema.
- Redirigere tale lista all'interno file **/exam/exercise5/socket.unit**

Exercise 6: Bash script

- Create uno script bash chiamato **/exam/exercise6/number.sh** con i seguenti dettagli:
    - Se lanciato passando come argomento un numero più grande di 100, dovrà restituire l'output "**grather then 100**"
    - Se lanciato passando come argomento un numero minore uguale di 100, dovrà restituire l'output "**OK**"
    - Se lanciato con qualunque altro argomento che non sia un numero dovrà restituire l'output "**not a number!!"**

Exercise 7: Bash script (**yum**)

- Create uno script bash sotto **/exam/exercise7** chiamato **manage-package.sh** con le seguenti caratteristiche:
    - Accetti in ingresso due parametri
        - il primo contenente il nome del package
        - l'azione da intraprendere
            - **search**: effettuerà una ricerca del possibile software da installare restituendo la lista trovata
            - **install**: installerà il software passato come argomento
            - **remove**: rimuoverà il software installato passato come argomento
            - **info**: restituirà le informazioni del software passato come argomento
    - Se non verranno passati argomenti o saranno inferiori o superiori a due, o non quelli permessi restituisca il messaggio "**Usage: manage-package.sh <packageName> <search/install/remove/info>"**
    - Si utilizzi una funzione per stampare a video il messaggio Usage.
    - **NB.** si utilizzi anche il case statement per la gestione dello script basato sulle azioni da intraprendere.

Exercise 8: HTTPD Dynamic content

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che HTTPD venga lanciato al boot della macchina e sia in ascolto sulla porta 80
- Aggiungere la regole firewall per poter accedere dall'esterno al servizio HTTPD
- 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/exercise8/
- Creare il file /exam/exercise8/index.php con il seguente contenuto

```php
<?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/index.php

Exercise 10: Docker

Install docker

- 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 Student" ogni 2 secondi, per un massimo di 5 volte, e poi uscire stampando la stringa "goodbye!!"
- Fare in modo che la parola Student possa essere modificata tramite variabile di ambiente passata allo start del container.

Question 1

- Quali sono secondo voi i principali benefici di una applicazione in container rispetto ad una applicazione installate sul sistema operativo.
    - Salvare la risposta sotto **/exam/question1**

Question 2

- Quali sono le funzioni presenti all'interno di un orchestratore di container come Kubernetes più vantaggiose rispetto ad una o più applicazioni in container gestite tramite ad esempio il tool Docker Compose
    - Salvare la risposta sotto **/exam/question2**