

## 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 pipeline, regular expression and command substitution

- Utilizzare il comando `find`, e l'utente `student`, per trovare tutti i files e directories sotto il path `/var` che contengono la lettera **a** o la lettera **b** o le lettera **c**. Per ogni file trovato determinare il file type utilizzando il comando `file` e redirigere l'output su `/exam/exercise1/find.txt`. Redirigere eventuali errori su `/exam/exercise1/error.txt`

## Exercise 2: Special file permission

- Utilizzare l'account **student**
- Creare sotto il path `/exam/exercise2/` le directory **directoryX** con **X** compreso tra 1 e 15
- all'interno di ogni directory sotto `/exam/exercise2/directoryX` creare i files
  - **exam2\_file(1..9)\_DATE.txt**
    - con **(1..0)**= tutti i numero tra 1 e 9
    - **DATE** la data di creazione file nel formato y-m-d (**date +%y-%m-%d**)

Il risultato sarà il seguente:

```
|-- exercise2/directory1
|   |-- exam2_file1_2022-02-08
|   |-- exam2_file2_2022-02-08
|   `-- exam2_file3_2022-02-08
etc...
|-- exercise1_directory2
|   |-- exam2_file1_2022-02-08
|   |-- exam2_file2_2022-02-08
|   `-- exam2_file3_2022-02-08
etc...
```

## Exercise 3: Default file permission

- Creare l'utente **exam2**, e fare in modo che nuovi file e directory creati dall'utente **exam2** alla creazione, di default non abbiano nessun permesso associato al **group owner** e gli utenti **others**.

## Exercise 4: User and Group

- Creare un nuovo gruppo chiamati **students** con GID 3005
- Creare gli utenti appartenenti al gruppo students **erossi** e **gbianchi**
  - l'utente **erossi** avrà le seguenti caratteristiche:
    - UID 3008
    - home directory /home/students/erossi
    - password erossi
    - dovrà cambiare password una volta ogni 30 giorni
  - l'utente **gbianchi** avrà le seguenti caratteristiche:
    - UID 3010
    - potrà accedere e gestire i file e directory appartenenti al gruppo **collaboration**
    - dovrà cambiare password al primo accesso
    - l'account scadrà dopo un anno dalla sua creazione

## Exercise 5: Shell environment and alias command

- Creare un nuovo comando o alias command a piacimento chiamato **userinfo** disponibile al login per **TUTTI** gli utenti del sistema. Chiamato dovrà stampare la seguente stringa  
user: <username> - working directory: <print working directory> - home\_directory:  
<home directory>

Esempio:

```
[student@hostname tmp]$ userinfo  
user: student - working directory: /tmp - home_directory:  
/home/student
```

## Exercise 5: Firewallld

- Abilitare in modo permanente l'accesso al sistema per i servizi sulla porta 8081/TCP

## Exercise 6: Bash script

Preparare uno script bash sotto **/exam/exercise6** chiamato **copyFile.sh** con le seguenti caratteristiche:

- prenda in ingresso come argomento una stringa **<string>** e una directory **<directory>**
- Lo script dovrà cercare tutti i file con estensione **.<stringa>** presenti all'interno della directory **<directory>** e in tutte le sue sotto directory
- Se non presente creare la directory **/exam/exercise6/copied\_files**
- Lo script dovrà copiare tutte le occorrenze trovate sotto la directory **<directory>** in **/exam/exercise6/copied\_files/**
- Verifiche:
  - lo script dovrà verificare che la directory passata esista. Se questa verifica fallisce dovrà uscire riportando in output il messaggio: **"error: directory <directory> doesn't exists"**
  - lo script dovrà verificare che gli argomenti passati in ingresso siano pari a due. Se questa verifica fallisce dovrà uscire riportando in output il messaggio **"usage: copyFile.sh <string> <directory>"**
- **ATTENZIONE:** si consiglia per la ricerca di utilizzare una directory non di sistema. Sconsiglio quindi **/etc /var /usr**. La cosa migliore sarebbe quella di utilizzare, ad esempio, directories e files creati durante l'esercizio 1. Il comando di copia file non è distruttivo, a meno che per qualche errore di concetto/distrazione non si inverta l'ordine di copia.
  - `cp --help`  
Usage: cp [OPTION]... [-T] **SOURCE DEST**  
or: cp [OPTION]... SOURCE... DIRECTORY  
or: cp [OPTION]... -t DIRECTORY SOURCE...  
Copy **SOURCE** to **DEST**, or multiple **SOURCE(s)** to **DIRECTORY**.

## Exercise 7: Docker

- La directory **/exam/exercise7** dovrà contenere i files:
  - **Dockerfile**
  - **start\_application.sh**
  - **docker-compose.yml**
- **Dockerfile** conterrà le istruzioni per la creazione della immagine Docker basata su fedora:lates con installato il servizio HTTPD e il servizio NGINX
- **entrypoint.sh** sarà lo script chiamato come ENTRYPOINT dalla immagine Docker, il cui compito è quello di andare ad attivare il processo HTTPD o il processo NGINX a seconda del valore riportato dalla variabile di ambiente SERVICE
  - se SERVICE=HTTPD allora dovrà essere attivato il processo httpd
  - se SERVICE=NGINX allora dovrà essere attivato il processo nginx
- **docker-compose.yml** verrà utilizzato per effettuare la build automatica della immagine (se non presente), per lanciare l'immagine Docker, per passare la variabile di ambiente SERVICE, per attivare port binding tra il vostro sistema di esame e la porta del servizio HTTP/NGINX presente sul container.
- **NGINX:**
  - porta di default è la porta 80 come per httpd
  - Per installare nginx potete utilizzare i repository configurati sul vostro sistema, così come fareste per installare httpd, quindi non è necessario cercare metodologie alternative.
  - Per attivare il servizio nginx (mai lanciato direttamente durante i laboratori proposti durante l'anno) in foreground utilizzare il comando `nginx -g, "daemon off;"`

## Question :

- Si riporti un esempio di oggetto deployment tramite file yaml (exam2.yaml da creare nel path /exam/question/exam2.yaml), supponendo di dover gestire una applicazione basata sulla immagine "exam/applicationexam:latest" con lable exam1 su 3 repliche attive.
- Cosa introduce secondo voi questo oggetto di interessante riguardo il deploy di una applicazione in kubernetes?