

## 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: Regular Expressions

- Usare l'utente **student** per creare il file `/exam/exercise1/text` contenente l'elenco di tutti i file e directory presenti sotto `/usr` utilizzando il comando "**ls -r1**"
- riportare in `/exam/exercise1/regex` la regular expression necessaria per ottenere a partire dal file `/exam/exercise1/text` tutte le occorrenze che facciano match con:
  - inizi con un carattere compreso tra il carattere minuscolo **a**, e il carattere minuscolo **m** (a b c d e ... m)
  - finisca con il carattere **a** oppure **b** oppure **t** oppure **n**
- Potete testare la vostra regular expression con il comando **grep** (grep regex `/exam/exercise1/text`)

## Exercise 2: Special file permission and users

- Creare una directory sotto `/exam/exercise3/students` dove gli utenti del gruppo **students**, **mark nadine** e **martin** potranno condividere files
- Solamente gli utenti del gruppo **students** potranno creare e cancellare i file sotto `/exam/exercise3/students`
- Tutti i file creati sotto la directory `/exam/exercise3/students` dovranno essere assegnati automaticamente al gruppo **students**

## Exercise 3: Default file permission

- Creare l'utente **bill** appartenente al gruppo **microsoft**.
- Fare in modo che nuovi file e directory creati dall'utente **bill**, di default (al login quindi), non possano essere letti scritti o visti, da nessuno al di fuori dell'utente **bill**.

## Exercise 4: HTTPD Dynamic content

- 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 **8081**
- Aggiungere la regole firewall per poter accedere dall'esterno al servizio HTTPD sulla porta indicata
- Fare in modo che la Document Root impostata per il vostro servizio sia `/exam/exercise4`
- creare il file `/exam/exercise4/index.html` contenente la stringa "Hello Exam!"
- 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:8081`

## Exercise 5: Bash script

- Create uno script bash sotto **/exam/exercise5** chiamato **exercise5.sh** con le seguenti caratteristiche:
  - accetti in ingresso un lista di argomenti a piacere
  - lo script dovrà stampare in output:
    - la lista dei parametri passati in ingresso
    - il numero di parametri passati in ingresso
    - per ogni entry passata contare il numero di caratteri per ogni parametro e riportarlo in output
    - Se non viene passato nessun argomento in ingresso riporti un errore generico a piacere

### **example:**

```
# bash /exam/exercise5/exercise5.sh minnie goofy mickey donald
Parametri passati: minnie goofy mickey donald
Numero di parametri passati: 4
Il parametro minnie contiene 6 caratteri
Il parametro goofy contiene 5 caratteri
Il parametro mickey contiene 6 caratteri
Il parametro donald contiene 6 caratteri
```

## Exercise 6: docker-compose

- La directory **/exam/exercise6** dovrà contenere i files:
  - **bin/Dockerfile**
  - **bin/entrypoint.sh**
  - **docker-compose.yml**
- **Dockerfile** il compito del container sarà quello istanziare una immagine con immagine di base **cavatortaluca/exam:centos8** senza nessun servizio installato.
- **user.sh** sarà lo script bash che avrà in carico la creazione dell'utente in fase di start del container come ENTRYPOINT. Prevedere anche un comando per lasciare un processo attivo in modo da lasciare il container up and running (ad esempio sleep 10000)
- **NB:** Il container dovrà prevedere sia ENTRYPOINT che CMD entry. Allo start del container tramite docker-compose andrete a fare l'override del CMD passando l'utente da docker-compose file, per cui prevedere un CMD default all'interno del Dockerfile a piacere
- **docker-compose.yml** dovrà gestire la build del container e il passaggio dello username tramite entry **command**

```
example_service:  
  [..] build entry  
  command: ["user1"]
```

Questo esempio andrà a fare override del CMD passando "user1" come argomento allo script definito come ENTRYPOINT

## Question :

- Applicazioni installate su sistema operativo standard, e in container. Quali le differenze principali.