

## 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 and regular expression

- Utilizzare il comando **find**, e l'utente **student**, per trovare sotto la directory / tutti i file che terminano con la stringa "**config**".
  - Redirigere lo standard output sul file **/exam/exercise1/find.txt**.
  - Redirigere lo standard error sul file **/exam/exercise1/find\_error.txt**.

## Exercise 2: File permission and Environment

- Creare un nuovo comando sotto **/exam/exercise2** chiamato **count**.
- **count** dovrà eseguire il seguente comando: **find . -type f | wc -l**
- Il proprietario di **count** sarà l'utente **root**
- Dovrà essere richiamato da tutti gli utenti del sistema come comando senza specificare il path assoluto
- Fare in modo che chiunque lanci il comando **count**, lo faccia con le grant dell'utente **root** e non con quelle dell'utente che lancia il comando.

Example:

```
[root@hostname ~]# count  
8
```

## Exercise 3: User and Group

- Creare il nuovo gruppo chiamato **Marvel** con GID 2000
- Creare gli utenti appartenenti al gruppo **Marvel** **thor**, **wolverine** e **hulk**
  - l'utente **thor** avrà le seguenti caratteristiche:
    - UID 2001
    - home directory **/home/marvel/thor**
    - dovrà cambiare password una volta al mese
  - l'utente **wolverine** avrà le seguenti caratteristiche:
    - UID 2002
    - potrà accedere e gestire i file e directory appartenenti al gruppo **x-men**
    - password **wolverine**
    - dovrà cambiare password al primo accesso e successivamente una volta ogni 3 mesi
    - l'account scadrà dopo tre anni dalla sua creazione
  - l'utente **hulk** avrà le seguenti caratteristiche:
    - UID 2010
    - home directory **/home/marvel/hulk**
    - password **hulk**
    - l'utente non potrà fare accesso al sistema

## Exercise 4: File permission

- Creare gli utenti **albert** e **manfred** appartenenti al gruppo **students**.
  - Fare in modo che tutti i nuovi files e directory creati dall'utente **albert** possano essere letti e scritti dall'utente e gruppo **students** e che gli altri utenti non appartenenti al gruppo non possano accedere, eseguire, listare o scrivere su ciò che viene creato dall'utente **albert** (file o directory).
  - Fare in modo che tutti i nuovi files e directory creati dall'utente **manfred** alla creazione possano essere letti eseguiti e scritti solamente dall'utente stesso e nessun altro utente sul sistema possa accedervi.
  - Fare attenzione che questa configurazione sia permanente al logout/login degli utenti albert e manfred

## Exercise 5: Managing Files with Shell Expansion

- Creare la seguente struttura sotto le directory **/exam/exercise5/exercise5\_dirX/fileY** con **X** compreso tra 1 e 10 e **Y** compreso tra 10 e 20

```
exercise5_dir1
|
|----- file10.txt
|----- etc..
|----- file20.txt
exercise5_dir2
|
|----- file1.txt
etc...
```

## Exercise 6: Bash script

- Create uno script bash chiamato **/exam/exercise6/wait.sh** che accetti in ingresso un argomento:
  - L'argomento in ingresso dovrà essere un numero
  - Se nessun argomento verrà passato o non sarà un numero stampi una stringa di errore.
  - Se l'argomento in ingresso è un numero stampi una stringa di conferma
  - Il compito dello script sarà quello di attendere N secondi prima di uscire con un messaggio di completamento. N=<parametro passato in ingresso>

esempio:

```
[root@hostname ~]# ./wait.sh
no argument found
[root@hostname ~]# ./wait.sh ciao
ciao: not a number
[root@hostname ~]# ./wait.sh 3
3: ok is a number
Wait for 3 seconds
Completed!!
```

## Exercise 7: Bash script

- Create uno script bash chiamato **/exam/exercise7/case.sh** che accetti in ingresso un argomento:
  - se l'argomento passato in ingresso sarà 100 restituisca la stringa **"You got 1st prize"**
  - se l'argomento passato in ingresso sarà 101 restituisca la stringa **"You got 2st prize"**
  - se l'argomento passato in ingresso sarà 102 restituisca la stringa **"You got 3st prize"**
  - se l'argomento passato in ingresso sarà 103 restituisca la stringa **"You got 4st prize"**
  - Se lanciato con qualsiasi altro argomento dovrà restituire l'output **"Sorry, try for the next time"**

## Exercise 8: Httpd

- Configurare due virtual host per servire i contenuti presenti sotto le directory:
  - **/exam/exercise8/www**
  - **/exam/exercise8/www.exam5.com/www**
- Creare il file **/exam/exercise9/www/index.html** con contenuto **"Default httpd exam 02/07/2021"**  
Creare il file **/exam/exercise9/www.exam5.com/www/index.html** con contenuto **"www.exam5.com"**
- Configurare i due virtual host per fare in modo che il contenuto servito da URL: **www.exam5.com** e **exam5.com** sia **"www.exam5.com"**. Per ogni altra richiesta via http, il contenuto che il server dovrà erogare sarà **"Default httpd exam 02/07/2021"**
- Si aggiungano le regole firewall necessarie per connessione dall'esterno in maniera permanente.
- Per fare il test dalla vostra macchina in cloud aws dovrete inserire nel file **/etc/hosts** la entry  
**127.0.0.1 www.exam5.com exam5.com**  
esempio:  

```
# curl localhost
Default httpd exam 02/07/2021
# curl exam5.com
www.exam5.com
```
- Per fare il test dalla vostra macchina locale verso il sistema in cloud dovrete inserire nel file **hosts** del vostro sistema la entry

**IPPUBBLICO www.exam5.com exam5.com**

esempio:

```
sul sistema ec2-35-152-64-166.eu-south-1.compute.amazonaws.com
IPPUBBLICO=35.152.64.166
# curl 35.152.64.166
Default httpd exam 02/07/2021
# curl exam5.com
www.exam5.com
```

## Exercise 9: FirewallD

- Create una nuova regola firewall per il servizio http (in aggiunta a quella inserita nell'esercizio 8) che vada ad inserire nei log di sistema, per le connessioni verso il servizio **http** il prefisso **"Exercise9\_firewall\_log\_httpd"** per un massimo di 4 nuovi messaggi per secondo
- Aggiungere il servizio http,https per la zona **work** rendendo la regola permanente.
- Disabilitare al boot il servizio firewalld

## Exercise 10: Docker and Docker-compose

- Creare la propria immagine Docker basata su **centos:8** chiamandola **exam/exercise10:1.0**
- La directory **/exam/exercise10** dovrà contenere i files:
  - **Dockerfile**
  - **docker-compose.yml**
  - **hello.sh**
- **hello.sh** dovrà contenere lo script bash il cui compito sarà di stampare “hello **exam!!**”. La stringa **exam** dovrà poter essere sostituita passando una variabile di ambiente allo start del container.
- preparare il file **docker-compose.yml** per avviare il container tramite **docker-compose**.
  - la build della immagine deve essere gestita tramite la compose, definendola nel file **docker-compose.yml**.

esempio:

```
docker-compose up
[...]
Successfully built 27ae437e5975
Successfully tagged exam/exercise10:1.0
[...]
Creating root_hello_1 ... done
Attaching to root_hello_1
hello_1 | Hello exercise10
root_hello_1 exited with code 0
```

### Question:

Si descriva a parole come in kubernetes viene gestita la comunicazione tra POD attraverso la risorsa **Service**. Quali sono secondo voi i maggiori benefici nell'utilizzo dei Service.