

Connessione alla propria macchina

To access your instance:

1. Open an SSH client.
2. Locate your private key file (yourname-exam5.pem).
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:
chmod 400 yourname-exam5.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 "yourname-exam5.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

- Creare la seguente struttura sotto le directory **/exam/exercise1/exercise1_dirX/fileY** con **X** compreso tra 1 e 10 e **Y** compreso tra 10 e 20

```
exercise1_dir1
|
|----- file10.txt
|----- etc..
|----- file20.txt
exercise1_dir2
|
|----- file1.txt
etc...
```

Exercise 2: Managing pipeline and regular expression

- Utilizzare il comando **ls** o **find**, e l'utente **student**, per trovare tutti i file e directory sotto **/usr** che finiscono con il carattere **a** o il carattere **b** o il carattere **c**. Redirigere lo standard output sul file **/exam/exercise2/find.txt**.

Exercise 3: User and Group

- Create due nuovi gruppi **student** e **exam**
 - **student** con GID 4000
 - **exam** con GID 4001
- Creare gli utenti appartenenti al gruppo **exam**: **thomas** e **lucy**
 - l'utente **thomas** avrà le seguenti caratteristiche:
 - **UID 3000**
 - home directory **/home/exam/thomas**
 - password **thomas**
 - dovrà cambiare password una volta ogni 5 mesi
 - dovrà poter accedere a file e directory appartenenti al gruppo **student**
 - l'account scadrà dopo due anni dalla sua creazione
 - l'utente **lucy** avrà le seguenti caratteristiche:
 - **UID 3002**
 - home directory **/home/exam/lucy**
 - password **lucy**
 - dovrà avere come gruppo primario: **exam**
 - dovrà avere come gruppi secondari: **student** e **users**
 - dovrà avere **/bin/sh** come login shell al posto di **/bin/bash**

Exercise 4: File permission and Environment

- Creare un nuovo alias command chiamato **gstat**.
- **gstat** dovrà eseguire il seguente comando: **ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm.**
- Fare in modo che il comando **gstat**, sia disponibile per **tutti** gli utenti del sistema.

Exercise 5: HTTPD Dynamic content

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che HTTPD venga lanciato al boot della macchina
- Aggiungere la regola 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/exercise5/**
- Creare il file **/exam/exercise5/index.php** con il seguente contenuto

```
<?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 6: Bash script

- Create uno script bash chiamato **/exam/exercise6/sum.sh** con i seguenti comportamenti:
 - Se lanciato passando come argomento due o più numeri, dovrà restituire la loro somma
 - Se lanciato con qualunque altro argomento che non sia un numero dovrà restituire l'output **"not a number!!"**

ex:

```
bash /exam/exercise6/sum.sh 20 10 30
60
```

```
bash /exam/exercise6/sum.sh 20
20
```

```
bash /exam/exercise6/sum.sh hello 23 44
not a number
```

Exercise 7: Bash script

- Create uno script chiamato **/exam/exercise7/managefile.sh**
 - Lo script accetterà come primo argomento il nome di un file e come secondo quale azione lo script dovrà eseguire (**add/remove**)
 - Se chiamato con l'opzione **add** dovrà creare il file passato come argomento
 - Se chiamato con l'opzione **remove**, dovrà eliminare il filename passato come argomento
 - Se lo script viene chiamato senza argomenti dovrà tornare il messaggio **"Usage: /exam/exercise7/managefile.sh <filename> add/remove"**

example:

```
bash /exam/exercise7/managefile.sh file1 add
file1 created
bash /exam/exercise7/managefile.sh file1 remove
file1 removed
bash /exam/exercise7/managefile.sh filename
"Usage: /exam/exercise7/managefile.sh <filename> add/remove"
```

Exercise 8: Systemd and Firewall

- Installare sul vostro sistema il servizio vsftpd tramite yum
 - Il servizio deve essere attivato al boot della macchina
 - Individuare il **pid** del servizio tramite **systemctl**
 - verificare su quale porta il servizio è in listening con il comando

```
netstat -tulpen | grep <pid>
```

esempio:

```
netstat -tulpen | grep 12345
tcp        0      0 0.0.0.0:12345 0.0.0.0:12345 LISTEN      12345
```
- Aggiungere la porta appena individuata alle regole firewall per il protocollo **TCP** (ricordarsi di rendere permanente la regola firewall)
- potete verificare che il servizio sia raggiungibile dalla vostra postazione con il comando **telnet**:

```
telnet ec2-x-xxx-xxx-xxx.eu-central-1.compute.amazonaws.com <PortNumber>
Trying x.xxx.xxx.xxx...
Connected to ec2-x-xxx-xxx-xxx.eu-central-1.compute.amazonaws.com.
Escape character is '^]'.
220 (vsFTPd 3.0.2)
quit << per uscire digitare quit
221 Goodbye.
Connection closed by foreign host.
```

Exercise 9: Firewall: Configure port forwarding

- Create una regola di port forwarding sul vostro sistema in ingresso sulla porta 8081/tcp verso la porta 22/tcp.
- Potete testare la nuova regola connettendovi via ssh al vostro sistema sulla porta 8081 con l'opzione -p 8081 del comando ssh.

Exercise 10: Docker

Install docker e docker-compose

- Creare la propria immagine basata su **centos:7**
- La directory **/exam/exercise10** dovrà contenere i files:
 - **Dockerfile**
 - **dockerexam.sh**
- **dockerexam.sh** sarà lo script bash il cui compito sarà quelli di stampare "hello <name>!!" il parametro name dovrà essere una variabile di ambiente contenente di default il valore "exam"
- L'immagine dovrà chiamarsi `exam/myhelloexam:1.0`

Risultato:

```
docker run exam/myhelloexam:1.0
```

```
Hello Exam!!
```

```
docker run -e ENV_VAR_NAME="Luca" exam/myhelloexam:1.0
```

```
Hello Luca!!
```

Exercise 11: Docker compose

- Creare un mariadb service via docker-compose con le seguenti caratteristiche.
 - db root password: "exam5"
 - db password: student
 - db name: exercise11
 - Utilizzate un bind locale per i volumi del mariadb service
 - `/exam/exercise11/mariadb`
- Utilizzate le immagini ufficiali dal Docker Hub (hub.docker.com)
- Il file **docker-compose.yml** dovrà essere messo sotto **/exam/exercise11**

Question 1

- Quali sono secondo voi i principali benefici nella gestione di una applicazione basata sui soli container che interagiscono tra di loro, e la stessa all'interno di una soluzione in cluster come kubernetes?
 - Salvare la risposta sotto **/exam/question1/**