

Connessione alla propria macchina

To access your instance:

1. Open an SSH client.
2. Locate your private key file (yourname-exam2.pem).
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:
chmod 400 name-exam2.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 "name-exam2.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 **find**, e l'utente **root**, per trovare sotto la directory / tutti i file e directory che terminano con la stringa “**config**”. Redirigere lo standard output sul file **/exam/exercise2/find.txt**.

find example

```
[centos@ip-172-31-30-213 ~]$ find ./
./
./bash_logout
./bash_profile
[...]
```

```
./file1
```

```
./file2
```

find manual

```
[centos@ip-172-31-30-213 ~]$ man find
```

Exercise 3: User and Group

- Create due nuovi gruppi **student** e **exam**
 - **student** con GID 2000
 - **exam** con GID 2001
- Creare gli utenti appartenenti al gruppo **exam**: **greco**, **lucato** e **morini**
 - l'utente **greco** avrà le seguenti caratteristiche:
 - **UID 3000**
 - home directory **/home/exam/greco**
 - password greco
 - dovrà cambiare password una volta ogni 3 mesi
 - dovrà poter accedere a file e directory appartenenti al gruppo **student**
 - l'utente **lucato** avrà le seguenti caratteristiche:
 - **UID 3001**
 - password lucato
 - dovrà cambiare password al primo accesso e successivamente una volta al mese
 - l'account scadrà dopo un anno dalla sua creazione
 - l'utente **morini** avrà le seguenti caratteristiche:
 - **UID 3002**
 - home directory **/home/exam/morini**
 - password morini
 - 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 comando sotto **/exam/exercise4** chiamato **gstat**.
- **gstat** dovrà eseguire il seguente comando: **ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm**.
- Il proprietario di **gstat** sarà l'utente **root**
- Il file dovrà avere i permessi di lettura scrittura ed esecuzione così impostati:
rwxr-xr-x
- Fare in modo che chiunque lanci il comando **gstat**, lo faccia con le grant dell'utente **root** e non con quelle dell'utente che lancia il comando.

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 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/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/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!!"**

```
bash /exam/exercise6/number.sh 2000
greather then 100
bash /exam/exercise6/number.sh 20
OK
bash /exam/exercise6/number.sh ciao
not a number!!
```

Exercise 7: Bash script

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

example:

```
bash /exam/exercise7/createuser.sh testuser add
bash /exam/exercise7/createuser.sh testuser remove
bash /exam/exercise7/createuser.sh testuser
"Usage: /exam/exercise7/createuser.sh <username> add/remove"
```

Exercise 8: Systemd and FirewallD

- 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:<PortNumber>
0.0.0.0:*                               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: Firewallld

- Create una nuova regola firewall per il servizio http (in aggiunta a quella inserita nell'esercizio 5) che vada ad aggiungere ai log delle connessioni, verso il servizio http il prefisso "**Exercise9_log_httpd**" sulla **default** zone.
- Limitate il log ad un massimo di 2 nuovi messaggi per secondo

Exercise 10: Docker

Install docker e docker-compose

- yum install -y docker
- enable docker to start at boot (systemctl enable docker)
- avviare docker (systemctl start docker)
- installare docker-compose
 - curl -L "https://github.com/docker/compose/releases/download/1.25.1/docker-compose-\$(uname -s)-\$(uname -m)" -o /usr/local/bin/docker-compose
 - chmod +x /usr/local/bin/docker-compose
- Creare una nuova immagine Docker basata su centos:7 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.

Exercise 11: Docker compose

- Creare una applicazione wordpress mysql tramite docker-compose con le seguenti caratteristiche
 - porta **locale** wordpress 8081
 - DB_USER=student
 - DB_PASSWORD=stUdEnE2020
 - DB_NAME=exercise11
 - Utilizzate un bind locale per i volumi dati delle applicazioni sotto
 - /exam/exercise11/worpress per wordpress
 - /exam/exercise11/mysql per mysql
- Utilizzate le immagini ufficiali dal Docker Hub (**hub.docker.com**)
- Una volta attiva l'applicazione, si dovrà poter accedere dall'esterno sulla porta 8081 (regole firewall)
- 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/