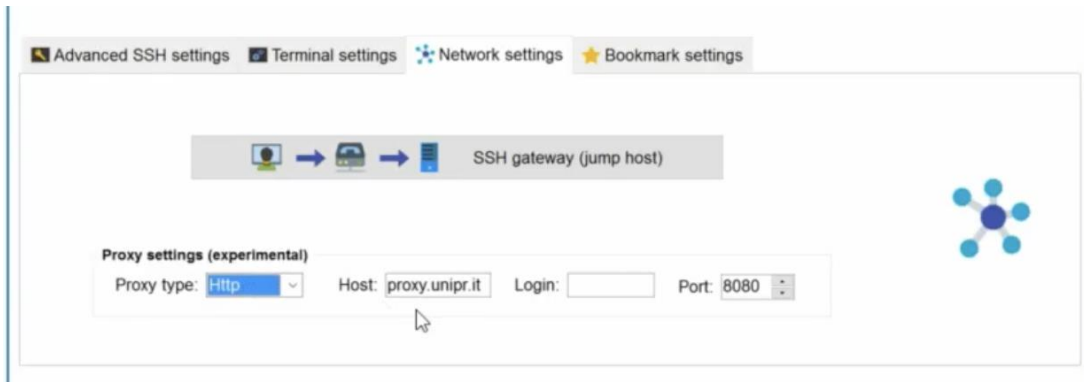
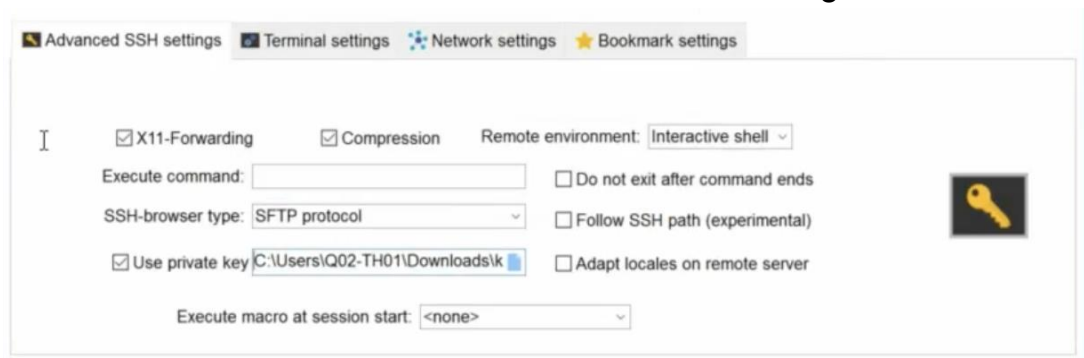


Connessione alla propria macchina

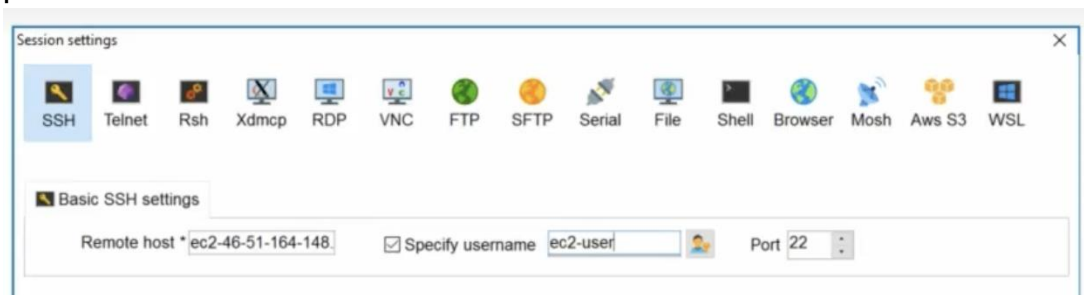
1. Aprire MobaXterm.
2. Tab session / SSH
3. Configurare il proxy dal tab Network Setting
 - a. Httpd
 - b. proxy.unipr.it
 - c. 8080



4. Passare la chiave SSH dal tab “Advanced SSH settings”



5. Inserire in Remote Hosts il dns della vostra istanza Linux
 - a. esempio: ec2-xx-xxx-xx-x.eu-central-1.compute.amazonaws.com
6. Username: **ec2-user**
7. port: **22**



Video accesso ai sistemi cloud dalle postazioni UNIPR:

[accesso_aws_pld_laboratori.mov](#)

Per passare all'utente **root** utilizzare il comando "**sudo su -**"
[ec2-user@ip-xxx-xx-xx-xxx ~]\$ **sudo su -**

Exercise 1: Regular Expressions

- Utilizzate il file **/exam/exercise1/text_file** per trovare tutte le occorrenze che:
 - contengono un **numero**
 - inizio riga con la parola **Listen**
 - finiscono con la parola **conf**
 - contengono il carattere **a** oppure **b**
- Riportare in **/exam/exercise1/regex_file** le 4 regular expression
- Potete testare la vostra regular expression con il comando **grep** (grep <regex> /exam/exercise1/text_file)

NB. (nelle quadre vuol dire caratteri a caso fra quelli proposti, per sovrascrivere un file ">", per appendere ">>")

```
mkdir /exam/exercise1
cd /exam/exercise1
touch text_file
echo "[0-9]" > exam/exercise1/regex_file
echo "^Listen" >> exam/exercise1/regex_file
echo "conf$" >> exam/exercise1/regex_file
echo "[ab]" >> exam/exercise1/regex_file
grep -E -f regex_file text_file
```

Exercise 2: Alias command and environment

- Creare un alias command disponibile per tutti gli utenti che chiamato **p-path**.
 - chiamato da riga di comando dovrà stampare a video il contenuto della variabile di ambiente PATH
- Assicuratevi possa essere lanciato da **qualunque** utente del sistema
- l'alias dovrà essere reso permanente

example:

```
[student@server ~]$ p-path
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
[root@server /]# p-path
/root/.local/bin:/root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

```
group (???????????)
vim /etc/bashrc (qui dentro per assicurarci che sia leggibile da chiunque)
dentro ci si scrive: "alias p-path='echo $PATH'"
source /etc/bashrc (per rendere già disponibile l'alias)
```

Exercise 3: File system permission and user

- Creare l'utente **mark** appartenente al gruppo primario **exam**.
- Fare in modo che nuovi file e directory creati dall'utente **mark**, di default (RENDERE QUESTO COMPORTAMENTO PERMANENTE), non possano essere letti scritti o visti da **other** ma avranno pieno accesso per **owner** e **group**.:::
- l'utente dovrà poter accedere a file e directory appartenenti al gruppo **students**
- L'utente dovrà cambiare password al primo accesso
- L'utente dovrà cambiare password ogni 60 giorni
- La home directory dell'utente **/home/mark** dovrà contenere il permesso **SGID** definito per il gruppo **student** in modo che ogni file creato in quella directory appartenga di default al gruppo **student**.

NB. Modificate le permission della home directory prima o dopo la creazione dell'utente per soddisfare i requisiti richiesti

```
groupadd exam
useradd -g exam mark
chmod 2770 home/mark    (cambia permessi e da tutto a owner e group)
groupadd student
usermod -aG student mark    (aggiunge mark a student)
chage -d 0 mark    (cambia password al primo accesso)
chage -M 60 mark    (cambia password ogni 60 giorni)
chmod g+s home/mark    (aggiunge stid a student)
chage -l mark    (verifica permessi di mark)
```

Exercise 4: HTTPD

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che HTTPD possa servire contenuti dinamici in PHP e che venga lanciato al boot della macchina
- Il servizio sarà in ascolto sulla **SOLA** porta **8090**
- Aggiungere la regole **firewall** per poter accedere dall'esterno al servizio sulla porta indicata in modo permanente
- Fare in modo che la Document Root impostata per il vostro servizio sia **/exam/exercise4**
- creare il file **/exam/exercise4/index.php** contenente lo script php

```
<?php
    echo "Hello Exam!!";
?>
```
- Potete verificare che il tutto funzioni tramite curl `http://localhost:8090`

```
[root@ip-172-31-22-69 ~]# curl http://localhost:8090
Hello Exam!!
```


```
dnf install httpd
dnf install php php_mysqlnd
vim /etc/httpd/conf/httpd.conf
Cercare "Listen" e metterlo su porta 8090
Assicurarsi che "LoadModule php_module" non sia commentato
Crea un file in "/etc/httpd/conf.d/" e scriverci dentro:
<VirtualHost *:8090>
    DocumentRoot "/exam/exercise4"
    ServerName localhost
</VirtualHost>
vim /exam/exercise4/index.php e scriverci dentro:
<?php
echo "Hello Exam!!";
?>
sudo firewall-cmd --zone=public --add-port=8090/tcp --permanent
sudo firewall-cmd --reload
systemctl start httpd
systemctl enable httpd
curl http://localhost:8090
```

Exercise 5: Bash script

- Create uno script bash sotto **/exam/exercise5** chiamato **cc.sh** con le seguenti caratteristiche:
 - accetti in ingresso due numeri e l'operatore matematico da utilizzare
 - lo script dovrà stampare in output:
 - la somma dei due numeri passati se l'operatore è: **sum**
 - la differenza dei due numeri passati se l'operatore è: **sott**
 - il prodotto dei due numeri passati se l'operatore è: **prod**
 - il quoziente dei due numeri passati se l'operatore è: **div**
 - verificare che il divisore non sia 0, altrimenti venga stampato un errore.
 - Se non vengono passati almeno tre parametri in ingresso riporti un errore generico a piacere
 - Se i parametri passati non sono numeri (ad eccezione dell'operatore) riporti un errore a piacere.

example:

```
bash /exam/exercise5/calc.sh 2 3 sum
5
bash /exam/exercise5/calc.sh 20 5 sub
15
ecc...
```



```
cd /exam/exercise5
vim cc.sh

#!/bin/bash

if [ "$#" -lt 3 ]; then
    echo "Almeno 3 parametri richiesti."
    exit 1
fi

num1=$1
num2=$2
operator=$3

if ! [[ $num1 =~ ^[0-9]+$ ]] || ! [[ $num2 =~ ^[0-9]+$ ]]; then
    echo "Non numerici."
    fi

case $operator in
    sum)
        result=$((num1 + num2))
        echo "$result"
        ;;
    sott)
        result=$((num1 - num2))
        echo "$result"
        ;;
    prod)
        result=$((num1 * num2))
        echo "$result"
        ;;
    div)
        if [[ $num2 -eq 0 ]]; then
            echo "Non si divide per 0"
            exit 1
        fi

        result=$((num1/num2))
        echo "Il prodotto è: $result"
        ;;
    *)
        echo "Operatore non valido. Utilizzare 'sum', 'sott', 'prod' o 'div'."
        exit 1
        ;;
esac

exit 0
```

Exercise 6: docker-compose

- Il servizio tramite **docker compose** dovrà gestire una applicazioni web costituita da un servizio *HTTPD*
- Sarà un servizio Apache installato su immagine di base **fedora:latest** o una immagine a piacimento, in ascolto sulla **porta 80** che andrà a montare sulla DocumentRoot il contenuto di una directory presente sotto **/exam/exercise6/content**
- Se sotto **/exam/exercise6/content** non è presente nessun file dovrà essere gestito tramite script di Entrypoint (oltre allo start di HTTPD) la creazione della index.html contenente la data di start del container. Se il file index.html è già stato creato non dovrà essere toccato o sovrascritto.
- **docker-compose.yml** verrà utilizzato per gestire start e la build della immagine, il mount tramite bind, e la gestione di un bind sulla porta locale 8000 e la porta 80 del container.

```
mkdir /exercise6/content
cd exercise6
mkdir dockerfiles
cd dockerfiles
```

```
vim docker compose.yml
version: '3'

services:
  httpd:
    image: fedora:latest
    ports:
      - "8000:80"
    volumes:
      - ./content:/var/www/html
    entrypoint: /entrypoint.sh
```

```
vim entrypoint.sh

#!/bin/bash

if [ ! -f /var/www/html/index.html ]; then

    echo " <html> <head> <title> Welcome </title> </head> <body>
<h1>Welcome to my website</h1> <p>Container started on $(date)</p>
<body> </html>" > /var/www/html/index.html

fi

httpd-foreground
```

```
chmod u+x entrypoint.sh
docker-compose up
```


Question :

le risposte andranno messe sotto la directory **/exam/question/**

- Come secondo voi Docker cambia l'approccio alla gestione delle applicazioni rispetto ad una gestione delle stesse su sistema operativo.
- Cosa sono i POD kubernetes e che differenza c'è tra POD e Container