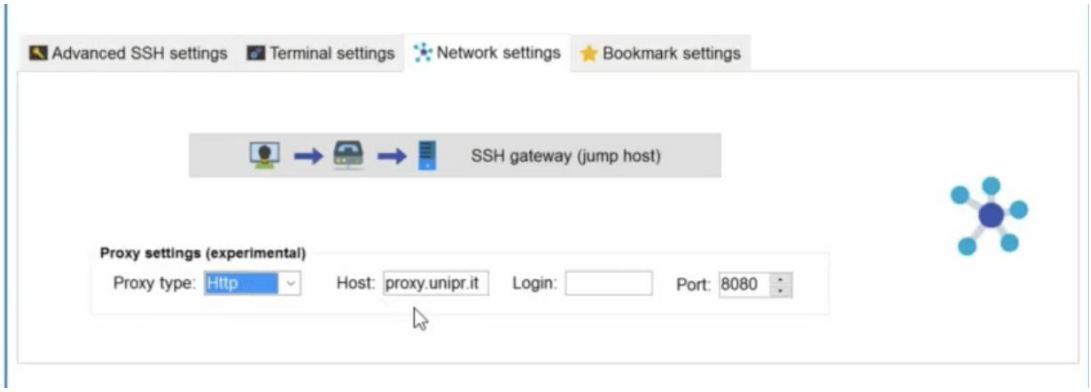
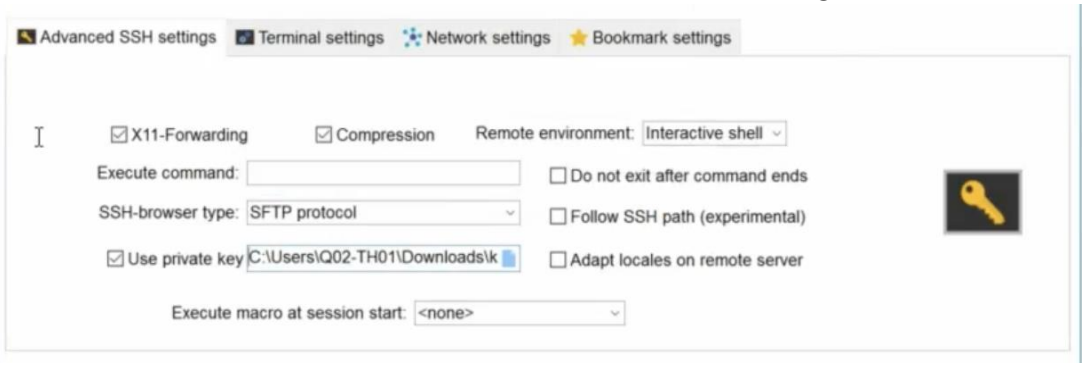
Connessione alla propria macchina

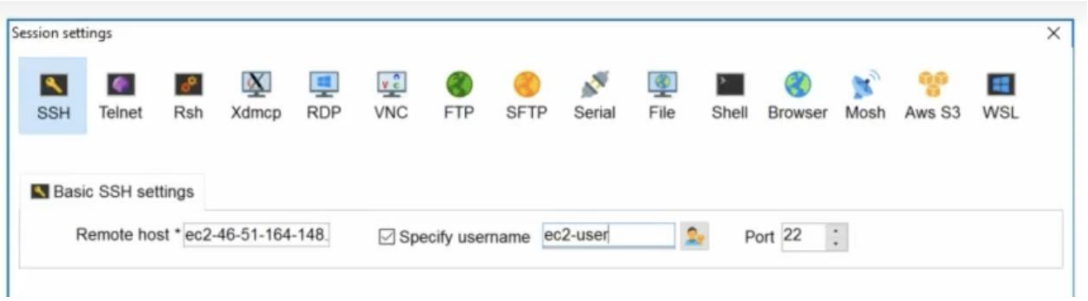
1. Aprire MobaXterm.
2. Tab session / SSH
3. Configurare il proxy dal tab Network Setting
    a. Httpd
    b. proxy.unipr.it
    c. 8080



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: Managing Files with Shell Expansion and Command substitution

- Utilizzare l'account utente **student**
- Creare sotto il path **/exam/exercise1/** le directory **exercise1_directoryX** con **X** compreso tra 1 e 200
- all'interno di ogni directory sotto **/exam/exercise1/exercise_directoryX** creare i files
    - **file(1..9)_HOSTNAME_DATE.txt**
        - con **(1..9)**= i numeri da 1 al 9
        - con **HOSTNAME** = hostname del sistema (**hostname -s**)
        - **DATE** la data di creazione file nel formato H:M:S

Il risultato sarà il seguente:

```
|-- exercise1_directory1
|   |-- file1_desktop_00:36:47.txt
|   |-- file2_desktop_00:36:47.txt
|   `-- file3_desktop_00:36:47.txt
etc...
|-- exercise1_directory2
|   |-- file1_desktop_00:36:47.txt
|   |-- file2_desktop_00:36:47.txt
|   `-- file3_desktop_00:36:47.txt
etc...
```

```
useradd student
su - student
cd /exam/exercise1
mkdir exercise1_directory{1..200}
touch ./exercise1_directory{1..200}/file{1..9}_$(hostname -s)_$(date
+"%H:%M:%S").txt
```

Exercise 2: Managing pipeline and regular expression

- utilizzando l'account utente **student** ricercare sotto la directory **/usr** tutti i nomi dei soli file presenti che iniziano con il carattere **a** oppure il carattere **b** oppure il carattere **c.**
    - Redirigere standard output sul file **/exam/exercise2/stdout.txt**.
    - Redirigere standard error sul file **/exam/exercise2/stderr.txt**.

```
su – student
find -regex '.*/[abc].*' > ./exercise2/stdout.txt 2>
./exercise2/stderr.txt
```

Exercise 3: User and Group

- Create due nuovi gruppi **teachers** e **students**
    - **teachers** con GID 3000
    - **students** con GID 3001
- Creare l'utente appartenente al gruppo **students**: **rossi**
    - L'utente **rossi** avrà le seguenti caratteristiche:
        - **UID** 3010
        - l'account scadrà dopo un anno dalla sua creazione
        - dovrà avere come gruppo secondario: **teachers**

su - root
groupadd -g 3000 teachers
groupadd -g 3001 students
useradd -u 3010 **-g** students **-G** teachers rossi
date -d "+365 days"
chage -E 25-03-11 rossi
chage -l rossi (per verificare stato utente)

Exercise 4: file permission

- Creare una directory sotto **/exam/exercise4** dove gli utenti che possono accedere al gruppo **students** potranno condividere files
- Tutti i file creati sotto la directory **/exam/exercise4** dovranno essere assegnati automaticamente al gruppo **students**

cd /exam/exercise4
chown root:students gesu
chmod 775 gesu
chmod g+s gesu

Exercise 5: Bash script

- Creare uno script bash sotto **/exam/exercise5** chiamato **manage-package.sh** con le seguenti caratteristiche:
    - Accetti in ingresso due parametri
        - il primo contenente il nome del package
        - il secondo l'azione da intraprendere
            - **search**: effettuerà una ricerca del possibile software da installare restituendo la lista trovata
            - **install**: installerà il software passato come argomento
            - **remove**: rimuove il software installato passato come argomento

- Se non verranno passati argomenti o saranno più o meno di due, o non quelli permessi restituisca il messaggio “**Usage: manage-package.sh <packageName> <search/install/remove/>”**
- Per le sole azioni **install remove** al termine, se andate a buon fine stampi a video il messaggio
  - **<packageName> installed/removed successfully**

  Se non completata con successo stampi a video il messaggio
  - **<packageName> install/remove error**

```
#!/bin/bash

package_name="$1"
action="$2"

if [ "$#" -ne 2 ]; then
	echo "Usage: manage-package.sh <packageName> <search/install/remove/>"
	exit 1
fi

case "$action" in
	"search")
		dnf search "$package_name"
		;;
	"install")
		sudo dnf install "$package_name"
		if [ $? -eq 0 ]; then
			echo "$package_name installed successfully"
		else
			echo "$package_name install error"
		fi
		;;
	"remove")
		sudo dnf remove "$package_name"
		if [ $? -eq 0 ]; then
			echo "$package_name removed successfully"
		else
			echo "$package_name remove error"
		fi
		;;
	*)
		echo "Invalid action"
		exit 1;
		;;
esac

exit 0
```

Exercise 6: docker-compose

- La directory **/exam/exercise6** dovrà contenere i seguenti files e directory:
  - **Dockerfile**
  - **entrypoint.sh**
  - **docker-compose.yml**
  - la directory **content**
- Il servizio tramite **docker compose** dovrà gestire l'applicazione hello-exam.
- **Dockerfile** conterrà le istruzioni per la gestione della vostra applicazione in container il cui servizio dovrà essere avviato tramite script di ENTRYPOINT.
  Nessun vincolo su immagine di base ecc….
- **entrypoint.sh** avrà il compito di scrivere sul file exam.txt il valore della variabile di ambiente EXAM per 6 volte e poi uscire
- **docker-compose.yml** verrà utilizzato per:
  - gestire start/build della immagine
  - inizializzare la variabile di ambiente EXAM con valore a piacere
  - gestire il bind locale con la directory /exam/exercise6/content che conterrà il file **exam.txt** popolato dallo script di ENTRYPOINT della applicazione in container

(siamo in /exam/exercise6)

touch Dockerfile entrypoint.sh docker-compose.yml
mkdir content

Dockerfile :

```
FROM ubuntu:latest

# Copia lo script di entrypoint
COPY entrypoint.sh /entrypoint.sh

# Imposta i permessi per rendere lo script di entrypoint eseguibile
RUN chmod +x /entrypoint.sh

# Imposta lo script di entrypoint come entrypoint per il container
ENTRYPOINT ["/entrypoint.sh"]
```

entrypoint.sh :

```
#!/bin/bash

export EXAM="valore_di_esempio"

for ((i=1; i<=6; i++)); do
  echo "$EXAM" >> /exam/exercise6/content/exam.txt
done

exit 0
```

docker-compose.yml :

```
version: '3'

services:
  hello-exam:
    build: .
    environment:
      - EXAM=valore_a_piacere
    volumes:
      - ./content:/exam/exercise6/content
```

sudo systemctl start docker (dentro /exam/exercise6)
docker build -t hello-exam .
docker run -v $(pwd)/content:/exam/exercise6/content hello-exam

Question :

- Si descriva a parole la differenza tra metodo dichiarativo e imperativo riportando un esempio (il più semplice che vi viene in mente) in BASH
- risposta sotto **/exam/question**