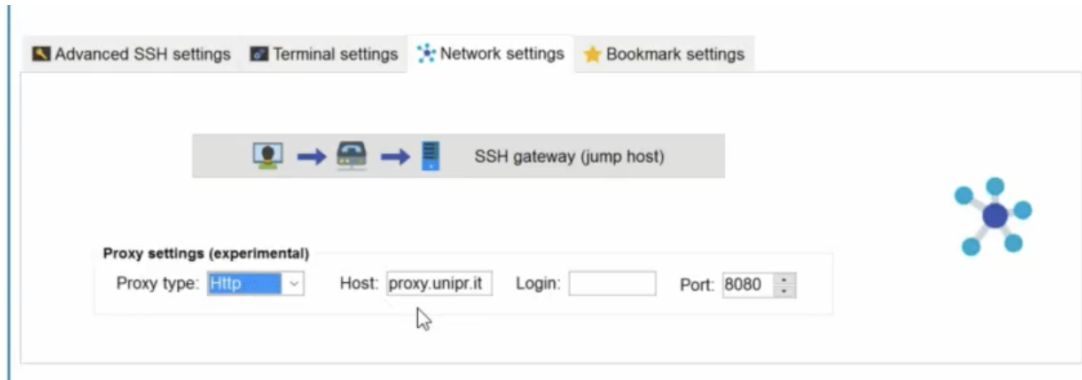
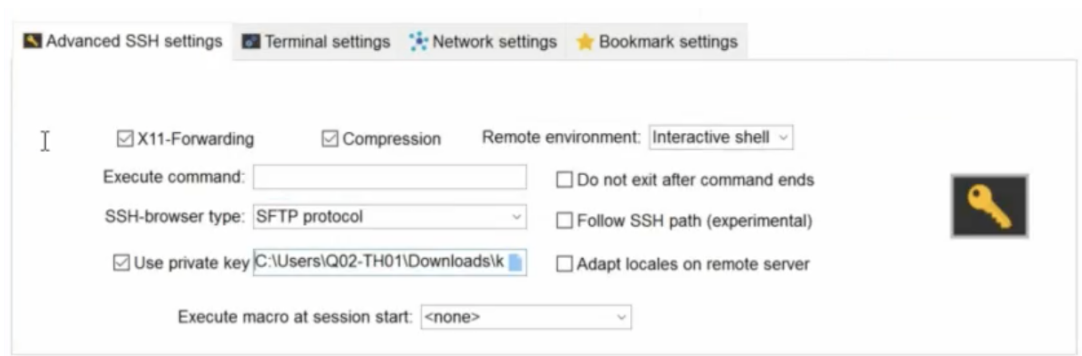


Connessione alla propria macchina

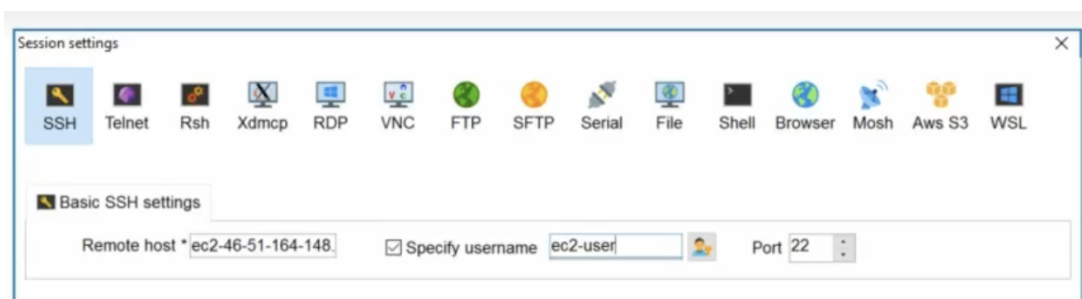
1. Aprire MobaXterm.
2. Tab session / SSH
3. Configurare il proxy dal tab Network Setting
 - a. Httpd
 - b. proxy.unipr.it
 - c. 8080



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

- Create il file **/exam/exercise1/text_file** contenente l'output del comando **"ls -l /"** per trovare tutte le occorrenze che:
 - contengano un **numero**
 - iniziano e finiscono con una **t**
 - finiscano con un numero
 - contengono il carattere **a** oppure **b** oppure **c** o combinazioni dei 3
- Riportare in **/exam/exercise1/regex_file** le regular expression
- Potete testare la vostra regular expression con il comando **grep** (grep <regex> /exam/exercise1/text_file)

Exercise 2: File permission and users

- Creare uno script (**NON UN ALIAS**) bash chiamato **user_name** che possa essere richiamato come comando senza specificare il path in cui si trova, e che stampi in output il nome dell'utente che lo lancia.
- Assicuratevi possa essere lanciato da **qualunque** utente del sistema

example:

```
[student@ip-172-31-35-174 ~]$ user_name
student
[root@ip-172-31-35-174 exercise1]# user_name
root
```

Exercise 3: Default file permission

- Creare l'utente **john** appartenente al gruppo **smith**.
- Fare in modo che nuovi file e directory creati dall'utente **john**, di default (al login quindi), non possano essere letti scritti o visti, da nessuno al di fuori dell'utente **john**.

Exercise 4: HTTPD

- Installare sul sistema il servizio HTTP/Apache
- Fare in modo che HTTPD venga lanciato al boot della macchina
- Il servizio sarà in ascolto sulla **SOLA** porta **8081**
- Aggiungere la regole firewall per poter accedere dall'esterno al servizio sulla porta indicata
- Fare in modo che la Document Root impostata per il vostro servizio sia **/exam/exercise4**
- creare il file **/exam/exercise4/index.html** contenente la stringa "Hello Exam!"
- Potete verificare che il tutto funzioni collegandosi 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:8081>

Exercise 5: Bash script

- Create uno script bash sotto **/exam/exercise5** chiamato **even_odd.sh** con le seguenti caratteristiche:
 - accetti in ingresso un lista di numeri
 - lo script dovrà stampare in output:
 - la lista dei soli numeri pari inseriti
 - la lista dei soli numeri dispari inseriti
 - Se non vengono passati almeno due parametri in ingresso riporti un errore generico a piacere
 - Se il parametro passato non è un numero riporti l'errore: ERROR only number please!

example:

```
bash /exam/exercise5/even_odd.sh 1 2 3 4 5
numeri pari: 2 4
numeri dispari: 1 3 5
```

Exercise 6: docker

- Creare una nuova immagine Docker basata su fedora 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.

Question :

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

- Quali sono le funzioni presenti all'interno di un orchestratore di container come Kubernetes più vantaggiose rispetto ad una o più applicazioni in container gestite tramite ad esempio il tool Docker Compose.
- Cosa si intende per *platform as a service* e perché possiamo considerare kubernetes un servizio di questo tipo.