



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

**(Laboratorio di)
Amministrazione di sistemi**

Predisposizione dell'ambiente per le esercitazioni

Marco Prandini

Dipartimento di Informatica – Scienza e Ingegneria

VM predisposte per il corso

- Verrà utilizzata la “box vagrant” Debian Bullseye ufficiale
- Durante le prime esercitazioni si mostrerà come predisporla in laboratorio
- Qui le indicazioni minimali per procedere in autonomia sul proprio computer - come detto il procedimento sarà chiarito a lezione:
 - Installare VirtualBox e Vagrant utilizzando il gestore software del proprio sistema operativo
 - Qualora fosse Ubuntu si ricorda che necessita l’abilitazione delle seguenti repository:
 - `sudo add-apt-repository universe`
 - `sudo add-apt-repository multiverse`
 - `sudo apt update`
 - Siti di riferimento:
 - <https://www.virtualbox.org/>
 - <https://www.vagrantup.com/>

Ambiente per le esercitazioni (1)

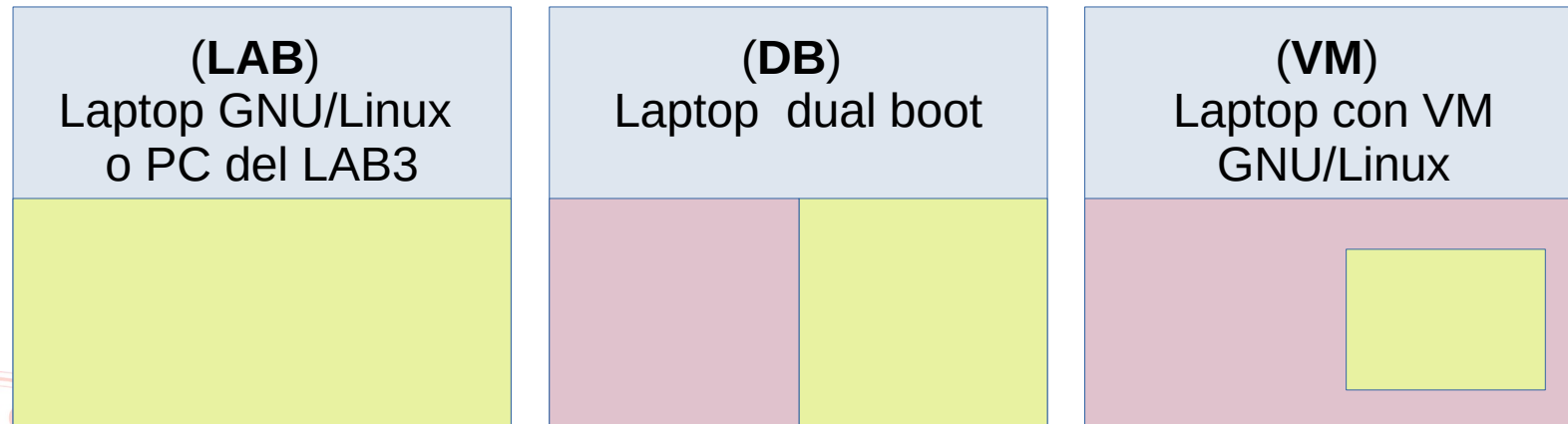
■ Finalità 1:

- Apprendere come realizzare semplici script bash

■ Requisito:

- Un qualsiasi sistema GNU/Linux

■ Tre soluzioni possibili



- Al fine di allineare didatticamente tutti gli studenti e le studentesse, chi fosse nella situazione (LAB) procederà come se dovesse realizzare la configurazione (VM)

Ambiente per le esercitazioni (2)

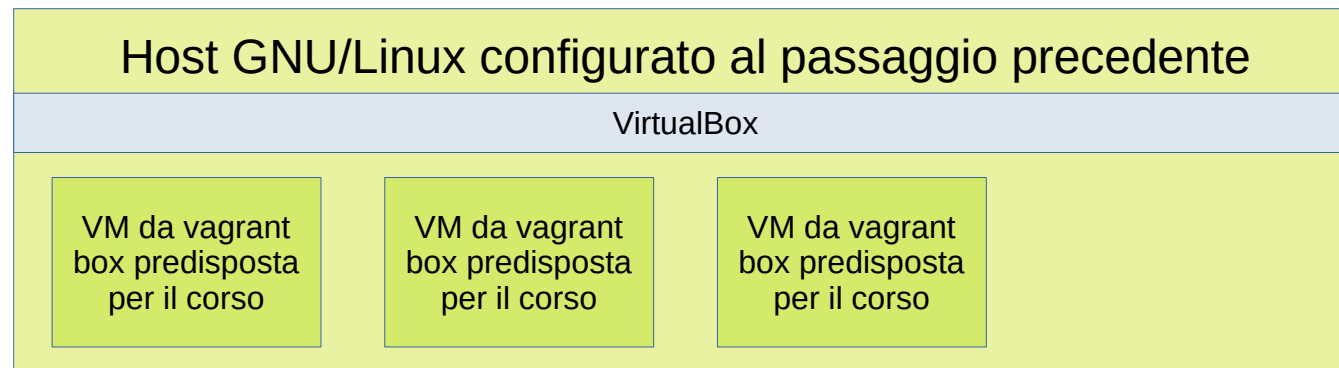
■ Finalità 2:

- Apprendere come configurare il sistema e i servizi principali di rete

■ Requisito:

- Più sistemi GNU/Linux uniformi, su cui operare come *root* e connessi in rete

■ Soluzione che adotteremo



■ Si noti come la configurazione (VM) descritta in precedenza porta a un funzionamento in **nested virtualization**

- Richiede specifica compatibilità hardware
- Ha prestazioni inferiori (in alcuni casi, significativamente)

Ambiente per le esercitazioni (3)

- Anche se imprecisa, per semplicità utilizzeremo la convenzione seguente per riferirci alle diverse macchine dello stack virtualizzato



Predisposizione per configurazione LAB

- VirtualBox dalla versione 4 conserva
 - le informazioni di base sulle VM registrate in `VirtualBox.xml`
 - Su linux è in `~/VirtualBox/` o in `~/config/VirtualBox/`
 - i dati delle diverse VM e le configurazioni specifiche in cartelle sotto `~/VirtualBox VMS/`
 - Per ogni VM un file *nome/nome.vbox*
- Queste informazioni sono utili per sistemare manualmente qualche dettaglio quando si vuol far funzionare un set di VM su più host (fattibile ma sconsigliato)
 - Per le esercitazioni è fondamentale modificare la configurazione del sistema perché nelle home del laboratorio non c'è spazio sufficiente per le VM
- NOTA: le istruzioni seguenti fanno riferimento a VirtualBox 6.1 in inglese su host Linux. Possono esserci piccole differenze rispetto ad altre combinazioni



Predisposizione per configurazione LAB

- L'host parte in modo testo. Per avviare l'interfaccia grafica digitare **startx**
- La prima volta che si usa il sistema di virtualizzazione
 - Andare nella cartella **large** e creare la cartella **VirtualBox VMs** (spazio incluso):
cd ~/large ; mkdir -p 'VirtualBox VMs'
 - Modificare il file **~/.config/VirtualBox/VirtualBox.xml** in particolare la riga che contiene: **defaultMachineFolder** inserendo il path alla nuova cartella creata dentro '**large**' es.
/home/LABS/s0000123456/large/VirtualBox VMs
 - OPPURE usare il comando (su un'unica riga):
vboxmanage setproperty machinefolder '/home/LABS/s0000123456/large/VirtualBox VMs'
 - Nella cartella '**large**' create la cartella '**.vagrant.d**' es.
cd ~/large ; mkdir -p .vagrant.d
 - Nella Home del vostro utente create un link simbolico verso la nuova cartella es.
ln -s ~/large/.vagrant.d ~/.vagrant.d

Predisposizione per configurazione LAB

- Creare una gerarchia di cartella dentro `‘.vagrant.d/boxes’` identica a quella presente dentro `‘/opt/vagrant/boxes’` es.

```
mkdir -p ~/large/.vagrant.d/boxes/debian-VAGRANTSLASH-bullseye64
```

```
mkdir -p ~/large/.vagrant.d/boxes/debian-VAGRANTSLASH-bullseye64/11.20221219.1
```

```
mkdir -p ~/large/.vagrant.d/boxes/debian-VAGRANTSLASH-bullseye64/11.20221219.1/virtualbox
```

- Creare i seguenti link simbolici:

- ```
ln -s /opt/vagrant/boxes/debian-VAGRANTSLASH-bullseye64/metadata_url \
```

```
~/large/.vagrant.d/boxes/debian-VAGRANTSLASH-bullseye64/metadata_url
```

```
ln -s
/opt/vagrant/boxes/debian-VAGRANTSLASH-bullseye64/11.20221219.1/virtualbox/*
\
```

```
~/large/.vagrant.d/boxes/debian-VAGRANTSLASH-bullseye64/11.20221219.1/
virtualbox/
```

- Facciamo un TEST !

```
cd ~/large ; mkdir -p test ; cd test
vagrant init debian/bullseye64
vagrant up
```



# Predisposizione per configurazione DB

## ■ Ridimensionare lo spazio dedicato a Windows per ricavare spazio non allocato

- Verrà usato per creare una partizione (o più) dedicata a Linux
- Nota: Windows non riconosce i tipi di partizione usati da Linux nè può accedere ai dati del filesystem ext4, ma non li “infastidisce”; Linux è in grado invece di accedere a filesystem di tipo FAT, FAT32, NTFS

## ■ Guida ufficiale Microsoft

<https://docs.microsoft.com/it-it/windows-server/storage/disk-management/shrink-a-basic-volume>

- Ricavare uno spazio non partizionato di circa 30GB, anche più se possibile



# Predisposizione per configurazione DB

- Scaricare un'immagine di un dispositivo avviabile con Linux
- Xubuntu come i pc del Laboratorio
  - Dal sito ufficiale <https://www.ubuntu-it.org/download/derive>
  - Versione dei LAB 20.04.5 a Febbraio 2023  
<https://cdimage.ubuntu.com/xubuntu/releases/20.04.5/release/xubuntu-20.04.5-desktop-amd64.iso>
  - Versione più leggera (per dimensioni e impatto su CPU e RAM) di Ubuntu, se non ci sono particolari vincoli si può usare una qualsiasi variante di Ubuntu



# Predisposizione per configurazione DB

- Il file scaricato ha estensione .iso ed è una copia bit per bit di quanto si troverebbe memorizzato acquistando il “DVD originale”, ormai storico
- Creare un dispositivo ottico o USB con tale immagine
  - Nota: deve essere fatto con un software apposito, non copiando il file ISO!
    - Si devono sovrascrivere boot sector, tabella delle partizioni, filesystem
    - Tutto il contenuto del pendrive USB o del DVD viene cancellato
  - Purtroppo richiede privilegi di amministrazione, quindi non può essere fatto direttamente dal PC del laboratorio, è necessario trasferire l'ISO sul proprio computer (si può usare lo stesso dispositivo USB che verrà poi cancellato installandoci l'immagine)
  - Da SUPER-HOST GNU/Linux:  
`sudo dd if=xubuntu-20.04.5-desktop-amd64.iso of=/dev/sdb bs=4M oflag=sync`
  - Da SUPER-HOST Windows: usare software come <https://www.balena.io/etcher/>



# Predisposizione per configurazione

- Installare VirtualBox da <https://www.virtualbox.org/wiki/Downloads>
- Se avete già un host ubuntu usare l'apposito pacchetto
  - si ricorda che necessita l'abilitazione delle seguenti repository:
    - `sudo add-apt-repository universe`
    - `sudo add-apt-repository multiverse`
    - `sudo apt update`
- <https://packages.ubuntu.com/jammy/virtualbox>
  - Scegliere la versione per il proprio sistema operativo host
  - Dopo l'installazione, è consigliabile scaricare dalla stessa pagina e installare l'Extension Pack
- Da fare anche per la configurazione DB
- In Lab è già installato

# Installazione di HOST

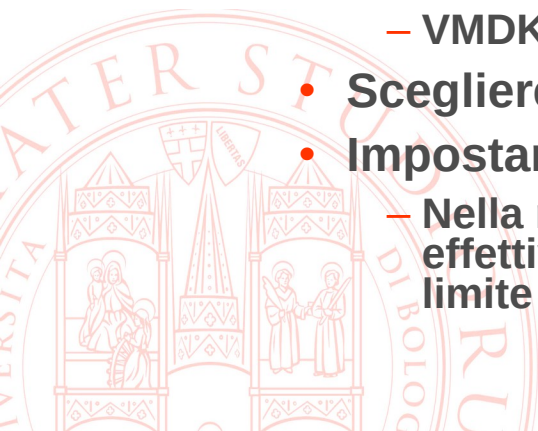
## ■ Per la configurazione DB

- Avviare il laptop con la memoria USB inserita
- Accedere al BIOS per pilotare l'avvio da USB

## ■ Per le configurazioni VM

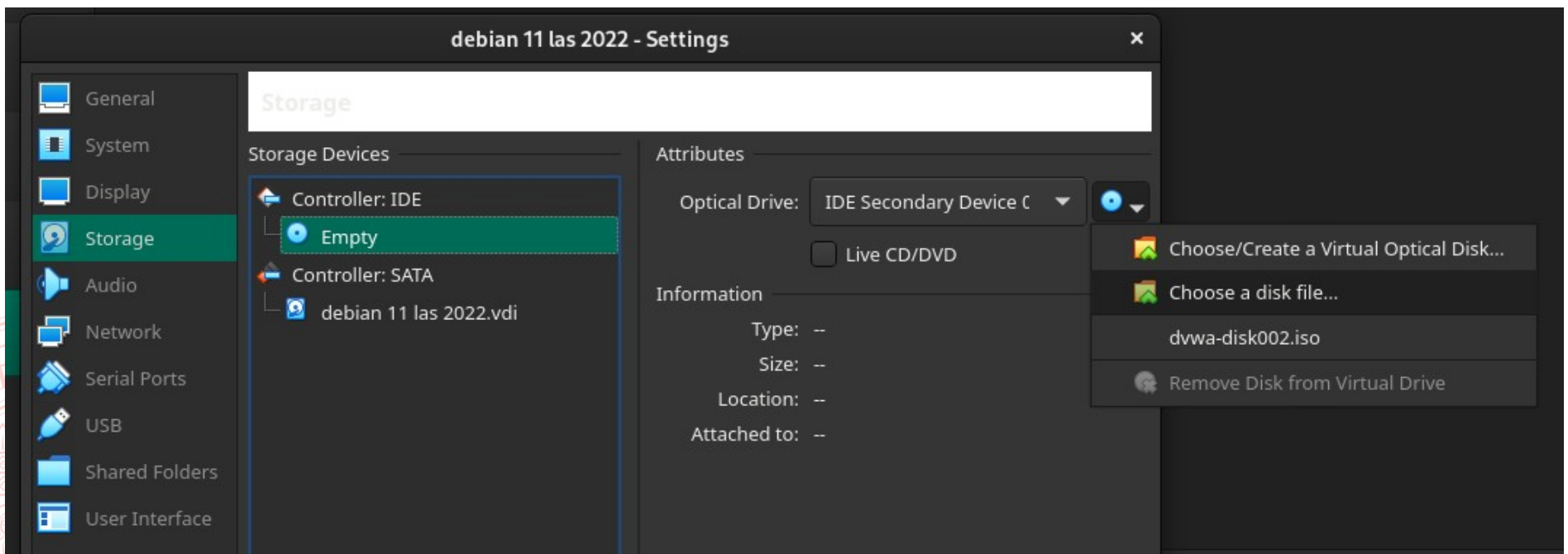
- Avviare VirtualBox
- Machine → New (Ctrl-N)
- Nella sequenza di schede che si presentano:
  - Scegliere un nome e impostare “Debian 64 bit” come versione
  - Impostare 2-4 GB come Memory size
  - Scegliere “Create virtual hard disk now”
  - Scegliere VDI come formato
    - VMDK se lo si vuole riutilizzare con VMWare
  - Scegliere “Dynamically allocated”
  - Impostare 30-40GB come spazio
    - Nella modalità “Dynamically allocated” verrà occupato solo lo spazio effettivamente utilizzato dalla VM, il numero impostato qui rappresenta il limite massimo (verrà visto come dimensione del disco dal guest)

Potrebbe  
essere  
necessario  
disattivare il  
Secure Boot



# Installazione di HOST (segue VM)

- Aprire le impostazioni della VM
  - Sezione Storage
  - Selezionare l'icona del CD sotto il "Controller: IDE"
  - Dall'icona del CD a destra di "Optical Drive" selezionare "Choose a disk file"
  - Scegliere il file **.iso** scaricato
- Avviare la VM



# Installazione di HOST

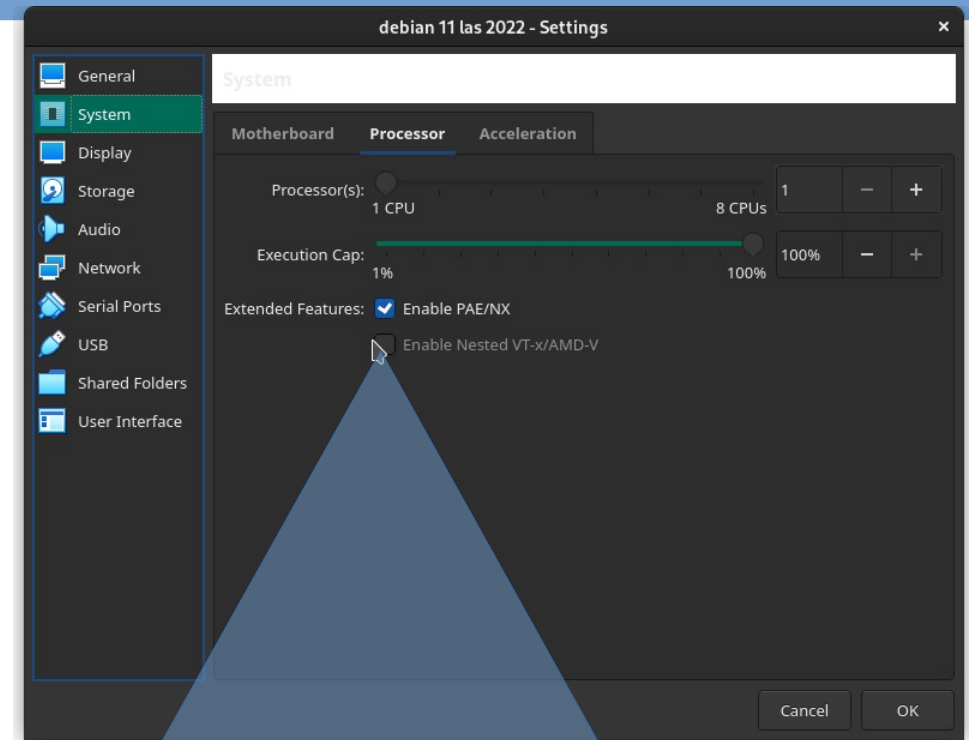
- Le slide seguenti dettagliano il processo per xUbuntu 20.04.5
- La procedura di installazione per xUbuntu è grafica e ancora più intuitiva, si dovrebbero molto facilmente individuare i passaggi equivalenti
  - Alla scelta del tipo di installazione si suggerisce:
    - minimale (no tool di office automation)
    - opzionale scaricare gli aggiornamenti durante l'installazione
    - utilizzare i driver proprietari





# Nested Virtualization (configurazione VM) VMWARE? WSL?

- Spegner la VM e prima di riaccenderla attivare la nested virtualization sul SUPERHOST
  - VT-x/AMD-V deve essere attivato a livello di BIOS
  - Impostazioni della VM
    - System
    - Processor
      - Enable PAE/NX
      - Enable Nested VT-x/AMD-V
      - Consigliato impostare 2 CPU



Se questa casella di VirtualBox non è selezionabile,  
aprire un prompt dei comandi Windows o un terminale  
Linux e digitare

**vboxmanage modifyvm "nome vm" --nested-hw-virt on**



# WSL

- Documentazione per [WSL](#)
- Installare Vagrant dentro WSL
- Configurare Vagrant dentro WSL per usare VirtualBox installato su Windows

```
echo 'export VAGRANT_WSL_ENABLE_WINDOWS_ACCESS="1"' >>
~/.bashrc
```

```
echo 'export PATH="${PATH}:/mnt/c/Program
Files/Oracle/VirtualBox"' >> ~/.bashrc
```

```
source ~/.bashrc
```

- Documentazione per Vagrant + WSL
- <https://developer.hashicorp.com/vagrant/docs/other/wsl>



# VM predisposte per il corso

- Installare VirtualBox Vagrant ed Ansible sul proprio computer:
  - `sudo apt install virtualbox vagrant ansible ansible-lint`
- Sul proprio computer a casa per la prima volta
  - `vagrant box add debian/bullseye64`  
*scarica l'immagine del sistema operativo*
- In tutte le configurazioni
  - `vagrant init debian/bullseye64`  
*crea il file di configurazione di default da fare solo al primo avvio*
  - `vagrant up`  
*avvia un'istanza della macchina virtuale*

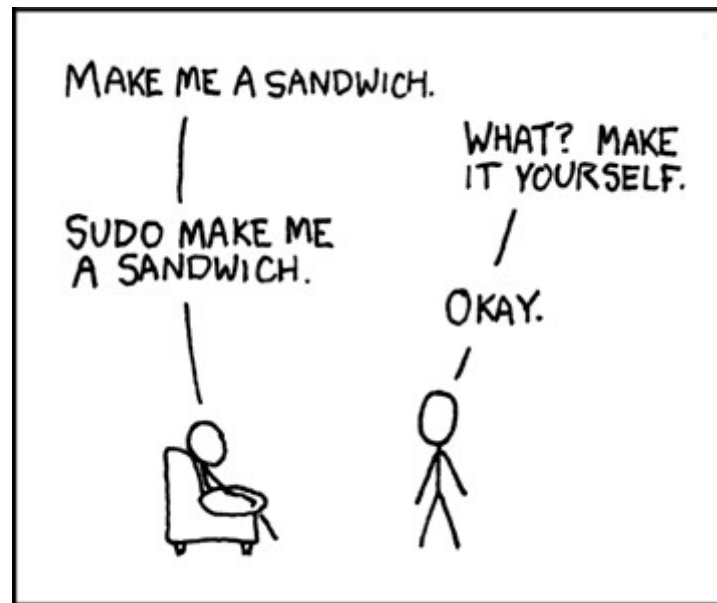


# Accesso

- Anche quando si lavora come sysadm, la quantità di operazioni da svolgere coi privilegi di root è molto limitata
  - è fortemente consigliabile utilizzare sempre l'account standard, che anche in caso di errori può provocare danni molto limitati
- Successivamente al login, è possibile cambiare identità
  - Il comando **su** (switch user) permette di "diventare" l'utente specificato come parametro, a patto di conoscerne la password
  - Senza parametri, equivale a **su root**
  - Consigliabile usare il parametro **-** per caricare l'ambiente dell'utente di destinazione (quindi il comando più comune sarà **su -** )
- Nota: sulle VM Vagrant
  - Si entra nella VM lanciando dalla stessa directory dell'host che contiene il Vagrantfile il comando **vagrant ssh**
  - l'utente root ha la password disattivata, quindi il metodo "**su -**" non è utilizzabile

# Accesso

- Oltre a **su**, esiste un modo migliore di utilizzare l'identità di root
  - **sudo** (super-user do) permette di eseguire un singolo comando come altri utenti (incluso root) senza conoscerne la password, a patto di essere stati preventivamente autorizzati



# sudo

- Sulle VM vagrant l'utente vagrant è già autorizzato a utilizzare sudo
  - Si lanciano comandi privilegiati con **sudo <comando>**
  - Si può diventare root con **sudo -i**
- Normalmente nelle distro di Linux l'utente scelto in fase di installazione è configurato allo stesso modo
- Se si desidera configurare **sudo**
  - Diventare root (con **su -** nel caso che root abbia una password valida e l'utente standard non sia ancora autorizzato a usare sudo, con **sudo -i** altrimenti)
  - Lanciare il comando **visudo**



# sudo

## ■ Due possibilità:

- Autorizzare l'utente “standard”, supponiamo si chiami **las**, a operare come qualsiasi utente, con la riga

**las ALL=(ALL:ALL) ALL**

- Autorizzare i membri del gruppo **sudo** a operare come qualsiasi utente, con la riga

**%sudo ALL=(ALL:ALL) ALL**

- Normalmente esiste già questa seconda riga nel file di configurazione di default
- In questo caso verificare che l'utente (es. **las**) sia nel gruppo **sudo** (comando **id**)
- Se non c'è, aggiungerlo (lanciare sempre come *root*)  
**usermod -aG sudo las**
- Efficace solo dal successivo login



# sudoers – qualche dettaglio in più

- Il file di configurazione di **sudo** è **/etc/sudoers**
- Leggete la man page... o anche no
  - forse l'unica che si sia mai vista che dice "don't despair..." dopo 200 righe di introduzione, prima di lanciarsi in 2600 ulteriori righe di delirio.
- Cose utili (grazie a <https://toroid.org/sudoers-syntax>)
  - formato delle direttive:  
**User Host = (Runas) Tag: Command**
  - da leggere come "User può eseguire Command coi privilegi di Runas su Host"
    - Runas si può specificare come *utente[:gruppo]* ; se manca, è implicito root (e nessun altro)
    - Tag: è facoltativo e può essere un'opzione che modifica alcuni aspetti dell'esecuzione
    - Host è utile per condividere lo stesso file *sudoers*, quindi la stessa security policy, su tutte le macchine dell'organizzazione, pur differenziandola
- Qualche esempio:
  - **%sudo ALL=(ALL:ALL) ALL**  
i membri del gruppo **sudo**  
su **qualsiasi host**  
possono eseguire **qualsiasi comando**  
a nome di **qualsiasi utente:gruppo**
  - **marco ALL= NOPASSWD:/bin/backup**  
*marco* può eseguire (solo come root) */bin/backup* senza digitare la propria password
  - **ams ALL=(ALL) sha224:IkotndXGTmZtH5ZNftRfIwkG0Wuiu0s7GoZ+6g== /bin/ls**  
*ams* può eseguire (come utente a sua scelta) */bin/ls* solo se il programma ha un digest corrispondente a quello memorizzato



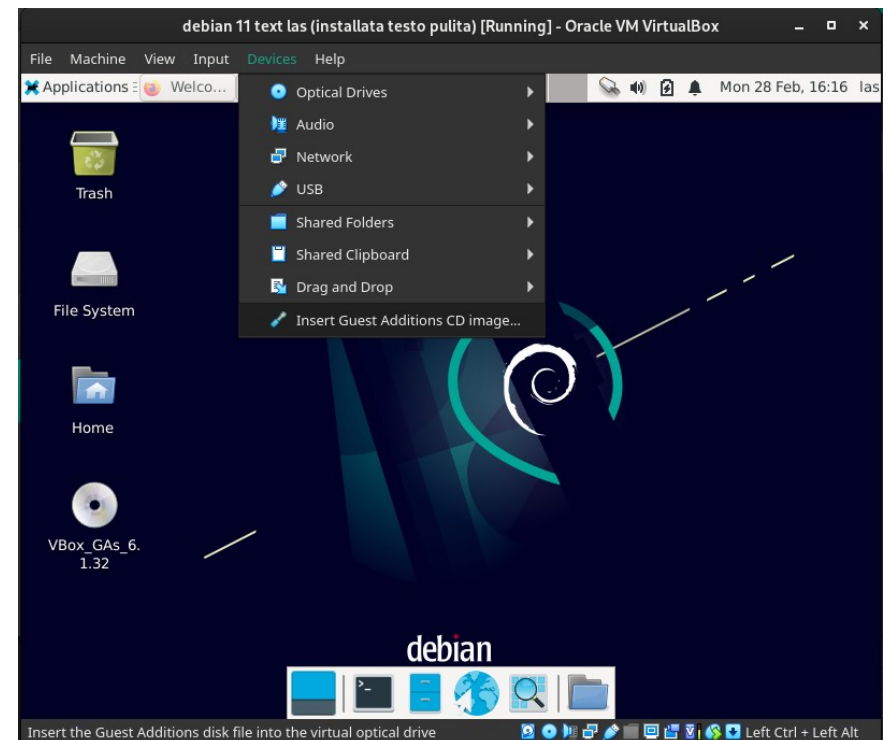
# Scambio file tra SUPER HOST e HOST

## Attraverso la cartella condivisa (1)

### ■ Preliminare: installare le *guest additions* nell'HOST

- Devices → Insert guest addition CD
- Aprire un terminale e diventare root
- Impartire:

```
mount /dev/cdrom /mnt
/mnt/VBoxLinuxAdditions.run
reboot
```

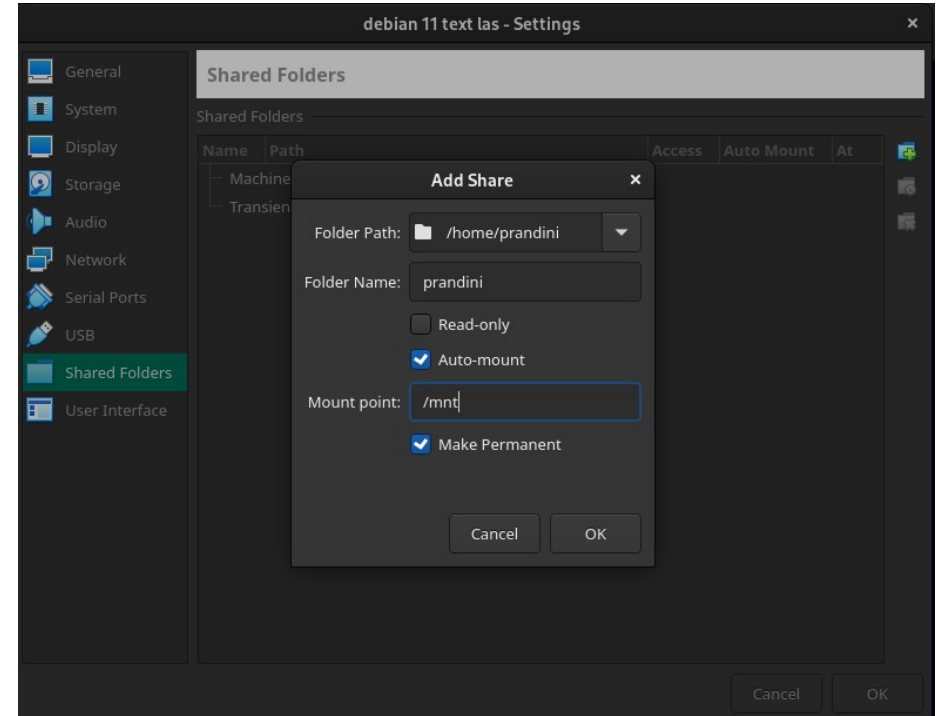
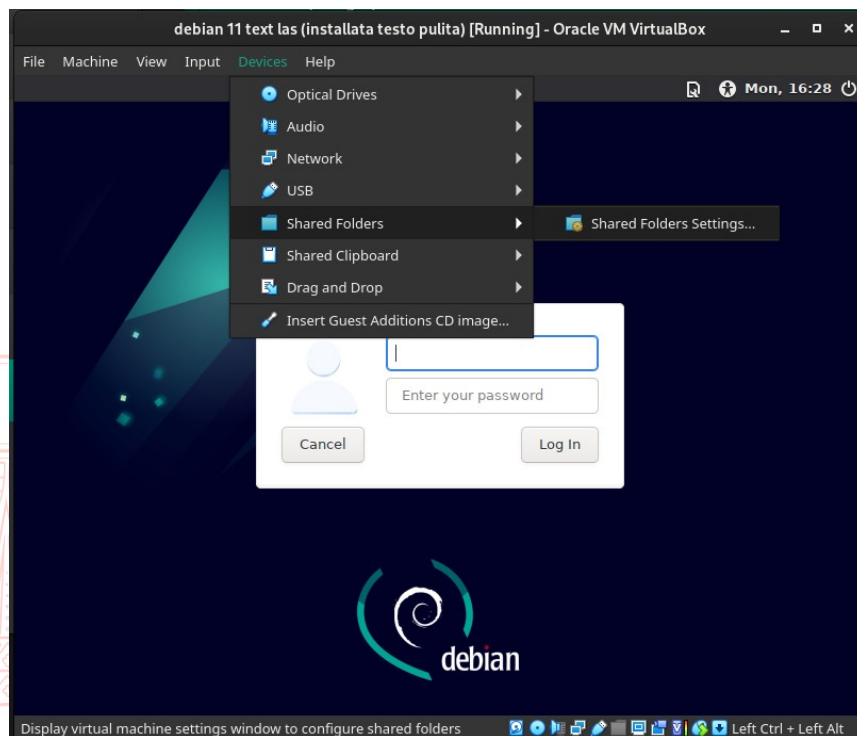




# Scambio file tra SUPER HOST e HOST

## Attraverso la cartella condivisa (2)

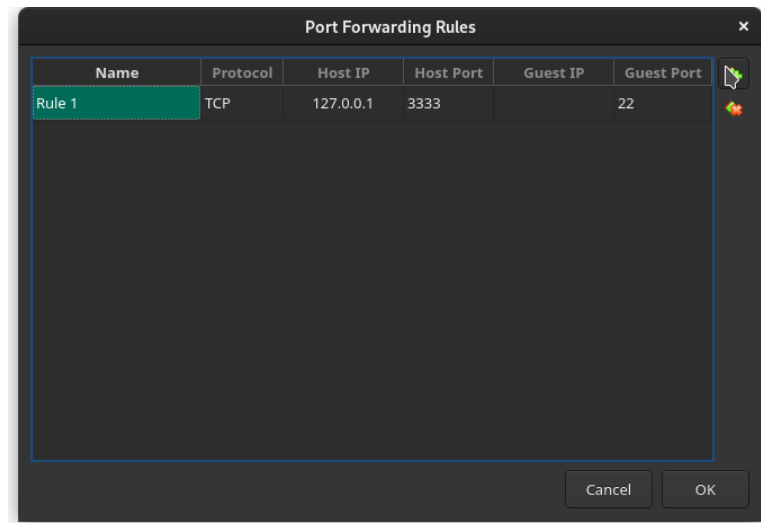
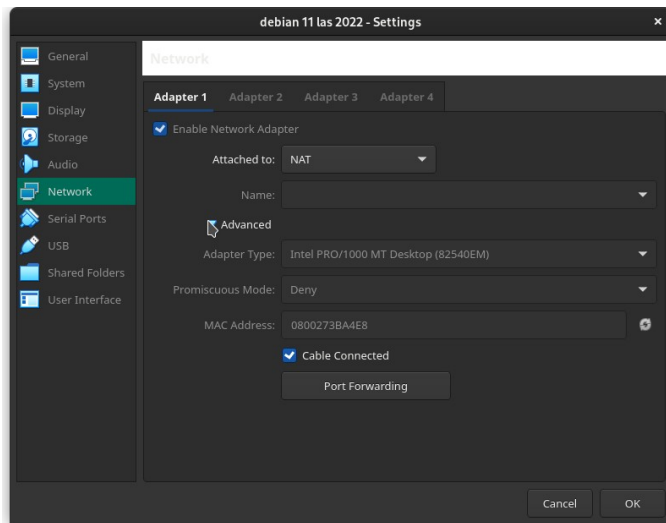
- Nelle impostazioni VirtualBox della VM HOST → Pannello *Cartelle condivise (Shared folders)*
  - Scegliere dal menu di esplorazione la cartella desiderata dell'host
  - Spuntare “Monta automaticamente” (Auto-mount) e “Rendi permanente” (Make Permanent)
  - Scrivere **/mnt** in “Mount point”



# Scambio file tra SUPER HOST e HOST

*Per mezzo della copia via rete*

- Configurare un port forwarding
  - Settings → Network → espandere Advanced → Port Forwarding
  - Tasto + → compilare i campi come in figura



- Nota per Ubuntu: SSH non è installato - eseguire come root in HOST

`apt install ssh`

- Comando su SUPERHOST per copia SUPERHOST → HOST

`scp source_file las@127.0.0.1:dest_path -p 3333`

- Comando su SUPERHOST per copia HOST → SUPERHOST

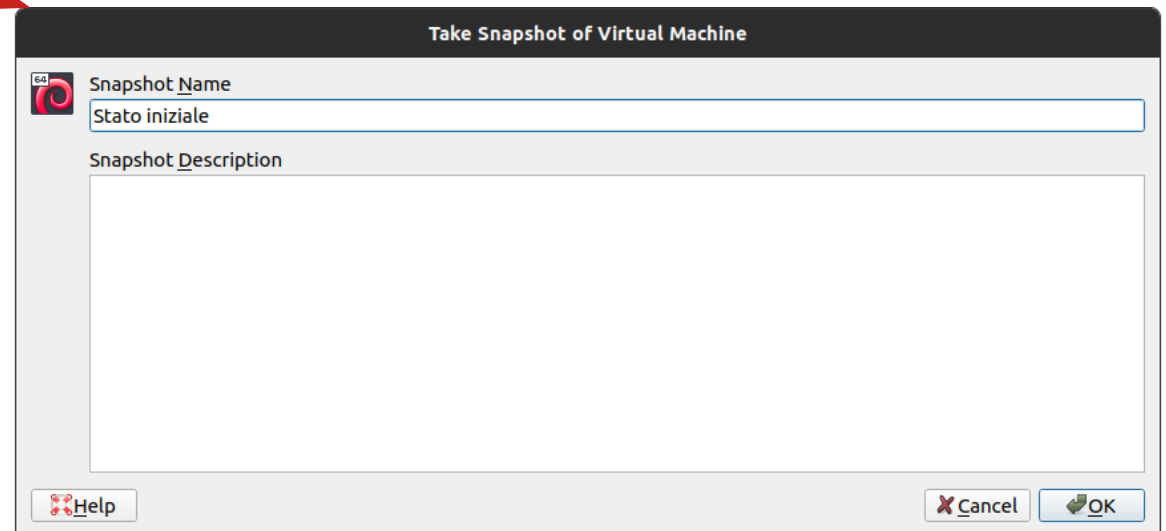
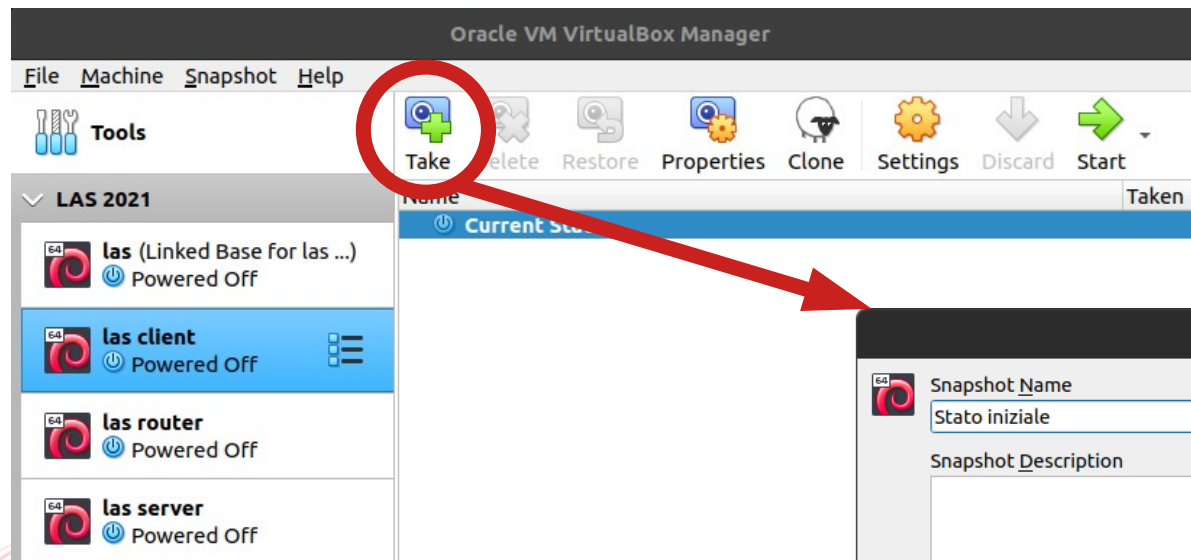
`scp las@127.0.0.1:source_file dest_path -p 3333`

# Shutdown

- Anche lo spegnimento di una macchina Linux (e unix in generale) richiede alcune operazioni.
- Si può abbandonare semplicemente la sessione di lavoro con `exit` (dalla shell di login) o con `logout`. Il sistema rimane comunque attivo e pronto ad altri accessi.
- Lo spegnimento richiede invece, di norma, l'accesso di root.
  - Sempre parlando di terminale. I sistemi grafici lo consentono all'utente.
  - Alcune installazioni permettono a chiunque si trovi in possesso della console (tastiera) di eseguire lo spegnimento (shutdown) premendo CTRL-ALT-CANC.
- root può invocare direttamente lo spegnimento, tipicamente con il comando:  
**shutdown [-h|-r] now**
  - **-h** indica la richiesta di arrestare il sistema (altrimenti: comando halt),
  - **-r** indica la richiesta di riavviare il sistema (altrimenti: comando reboot).
  - **now** indica quando eseguire l'operazione. E' possibile, ad esempio, lasciare agli utenti collegati alcuni minuti per terminare il proprio lavoro.

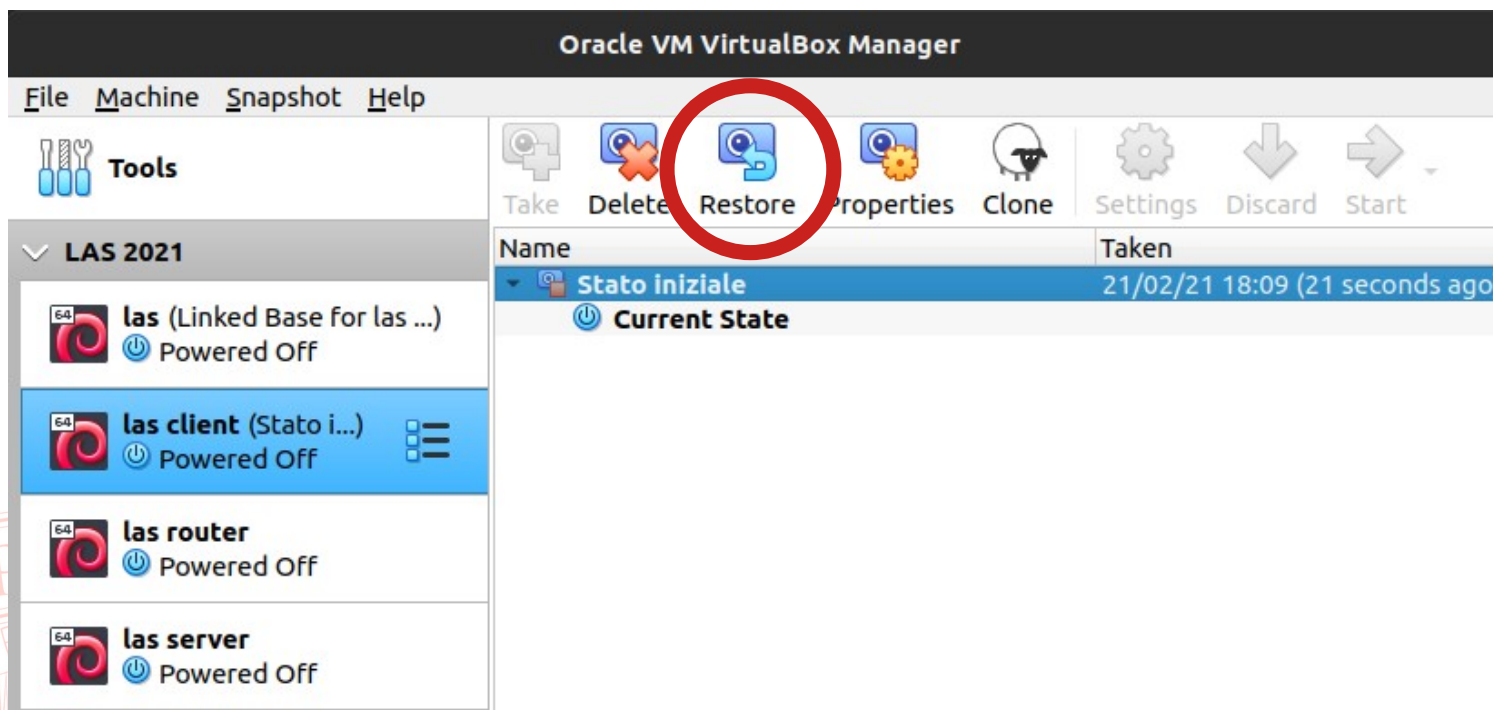
# Snapshot (della VM Host)

- È possibile congelare lo stato di una VM in modo da poterlo ripristinare in seguito a modifiche con esito indesiderato
- Selezionare una VM
- Con il tasto “Take” scattare uno snapshot e chiamarlo, ad esempio, “Stato iniziale”



# Recupero snapshot

- La macchina ora funzionerà usando il disco virtuale “Current State”, per ripristinarla a Stato Iniziale basta selezionarlo e cliccare sul tasto “Restore”



# Utenti Apple Silicon

- Possibili Virtualizzatori (noi non li supportiamo)
  - <https://www.vmware.com/it/products/fusion.html>
  - <https://www.parallels.com/it/>
- Possibilità di usare **HomeBrew** per installare Vagrant ed Ansible
  - <https://formulae.brew.sh/cask/vagrant>
  - <https://formulae.brew.sh/formula/ansible>
- Possibile box con Linux già configurata MA non è Debian sarà compito del utente trovare i comandi equivalenti qualora siano necessari
  - <https://app.vagrantup.com/almaLinux/boxes/9.aarch64>

