

# Manuale di Laboratorio

Laboratorio di Amministrazione di  
Sistemi - a.a. 2017/2018

## Le workstation

- Le stazioni di lavoro del laboratorio possono essere utilizzate in modalità Linux o Windows, scegliendo al boot
- Utilizzeremo sempre Linux
- Le modalità di login sono descritte nel materiale disponibile in laboratorio
- **RICORDATE DI SPEGNERE LE MACCHINE AL TERMINE DELL'USO**
  - Prima di tutto quelle virtuali (comando *halt* eseguito da root)

# Amministrazione e macchine

- Molte delle attività che svolgeremo
  - richiedono i privilegi di amministratore
    - modifica di permessi, invio di segnali, uso di funzioni di rete riservate, manipolazione dei filesystem, ...
  - possono di conseguenza causare danni
    - cancellazione di file di sistema, modifica di file di configurazione, ...
  - non sono quindi attuabili su macchine la cui stabilità deve essere garantita

---

## Virtualizzazione

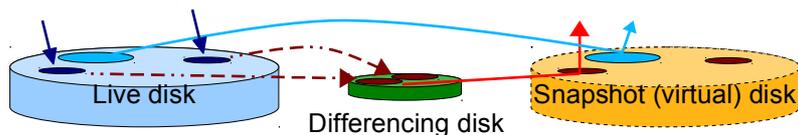
- Sistemi come VirtualBox, VMware, Xen, Qemu, UML, KVM lanciati come una normale applicazione in un sistema (*host*) emulano il funzionamento di un PC (o altro hardware), e permettono quindi l'esecuzione di un altro S.O. (*guest*)
- Utilizzeremo VirtualBox per lanciare macchine virtuali su cui lavorare con pieni privilegi
  - libertà d'azione
  - facilità di ripristino
  - portabilità sul proprio PC

# VirtualBox

- VirtualBox è un prodotto originariamente di Innotek, acquisita nel 2008 da Sun Microsystems, a sua volta acquisita da Oracle nel 2009.
- Già nel 2007, Innotek aveva reso open i sorgenti, cosicché per alcuni anni sono state disponibili due versioni del SW
  - Open Source Edition
    - Licenza GPL = nessuna restrizione d'uso
  - Closed Edition
    - Sorgenti non disponibili
    - Gratuita per PUEL (Personal Use / Evaluation License)
    - Gratuita per uso accademico
    - Feature avanzate rispetto a OSE
- Ora rilasciato con licenza GPLv2
  - Extension Pack ancora sotto PUEL

## COW e snapshots

- VirtualBox può avvalersi di snapshot per congelare lo stato di un disco virtuale
  - uso originale: backup coerenti a caldo con Copy On Write
  - creo un disco virtuale “snapshot”
  - lascio che i processi del S.O. lavorino sul disco originale
  - do l'illusione alla procedura di backup che legge dallo snapshot che nessuno stia modificando il disco

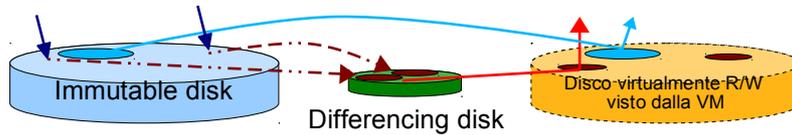


*ogni scrittura sul disco live provoca il salvataggio del settore originale sullo spazio di appoggio dello snapshot (differencing disk)*

- *i settori copiati vengono enumerati in una exception table*
- *la lettura sullo snapshot disk di un settore modificato viene fatta dal differencing disk*
- *la lettura sullo snapshot disk di un settore non modificato viene fatta dal live disk*

# COW e snapshots

- Uso per la condivisione dello storage tra più macchine virtuali simili
  - Installo una VM “base” su di un disco e lo dichiaro Immutabile
  - Utilizzo quel disco come device di altre VM



*Per ogni VM viene allocato un differencing disk di appoggio; ogni scrittura sul disco immutabile viene in realtà eseguita sul differencing disk*

- *i settori corrispondenti vengono enumerati in una exception table*
- *la lettura di un settore modificato viene fatta dal differencing disk*
- *la lettura di un settore non modificato viene fatta dall'immutable disk*

## Esercitazioni

- Le esercitazioni saranno svolte col supporto del tutor, dott. Andrea Melis
- Contatti:  
a.melis@unibo.it

# Utilizzo del LAB4

- Accesso alle postazioni
  - Gli studenti per usare i PC devono usare esclusivamente le credenziali personali di Ingegneria, ottenibili in due modi:
  - In laboratorio: selezionando dal menu che appare all'accensione dei PC la voce "Crea Account"
  - Via web: al sito [https://infoy.ing.unibo.it/new\\_account](https://infoy.ing.unibo.it/new_account)

---

# Utilizzo del LAB4

- Per ottenere le credenziali personali di ingegneria è necessario:
  - essere regolarmente iscritti
  - aver pagato la rata delle tasse da almeno 3 giorni
  - essere in possesso delle proprie credenziali di Ateneo (nome.cognome@studio.unibo.it oppure nome.cognome@unibo.it)

# Utilizzo del LAB4

- I laboratori sono spazi comuni, occorre il rispetto di alcune regole:
  - I tavoli devono essere lasciati puliti e le sedie al proprio posto
  - In laboratorio non si consuma cibo
  - Lo stato delle connessioni dei pc non va modificato neppure temporaneamente: non va scollegata alcuna periferica e nessun cavo, compresi quelli di alimentazione e di rete
  - Non è possibile collegare PC portatili all'impianto elettrico del laboratorio
  - Esiste la raccolta differenziata dei rifiuti

---

## VM predisposte per il corso

- La directory */materiali/prandini2017* contiene l'immagine appliance **LAS-debian87-base-NOUSB.ova**
- La procedura descritta di seguito predispone il sistema del laboratorio, ma esaminando quanto detto si può facilmente riprodurre sul proprio host
  - Anche Windows o MacOS X (le differenze nei file di configurazione dovrebbero essere facilmente trattabili)
  - Scaricando l'appliance dal link fornito sul sito del corso se non lo si è potuto fare in laboratorio

# VirtualBox Registry

- 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
  - Per le esercitazioni è fondamentale modificare la configurazione del sistema perché nelle home del laboratorio non c'è spazio sufficiente per le VM
  - Dettagli nella guida separata alla configurazione manuale

## Prima predisposizione (1)

- L'host parte in modo testo. Per avviare l'interfaccia grafica digitare **startx**
- La prima volta che si usa il sistema di virtualizzazione
  - Inserire una memoria USB da almeno 2GB, meglio 4
  - Verificare dove è stata montata (comando **df** da terminale), es **/media/s0000123456/MYDISK**
  - Creare una directory nella memoria esterna

```
mkdir /media/s0000123456/MYDISK/LAS.VM
```
  - Creare una subdirectory per le VM

```
mkdir '/media/s0000123456/MYDISK/LAS.VM/VirtualBox VMs'
```
  - Creare un link dove VirtualBox si aspetta di trovare le VM e lo spazio predisposto sulla memoria esterna

```
cd
rm -rf 'VirtualBox VMs'
ln -s '/media/s0000123456/MYDISK/LAS.VM/VirtualBox VMs'
```

## Prima predisposizione (2)

- Avviare VirtualBox (menu Applicazioni → Sistema)
- Creare la rete locale per la comunicazione host-VM
  - File → Preferences → Network
  - Se in elenco non c'è già una rete di nome **vboxnet0**, crearla col tasto “add” 
- Importare l'appliance
  - File → Import Appliance
  - Sfogliare fino a trovare /materiali/prandini2017/LAS-debian87-base-NOUSB.ova
  - **(Togliere la spunta dal controller USB nei settings se si è scaricata l'altra immagine)**

## Prima predisposizione (3)

- Non toccheremo mai questa VM, ma ne creeremo (almeno) 3 *linked clones* che occuperanno su disco solo lo spazio dovuto alle differenze
  - Right-click sulla VM **LAS-debian87-base** → Clone
  - Nome della nuova VM: **LAS-debian87-Client**
  - Spuntare la casella “Reinitialize the MAC address...”
    - (Next)
  - Selezionare “Linked Clone”
    - (Clone)
- Ripetere la procedura per
  - **LAS-debian87-Router**
  - **LAS-debian87-Server**

# Modalità di accesso

- Credenziali sulle VM
  - Utente **root** password **gennaio.marzo**
  - Utente **las** password **las**
- ***Ricordate che cliccando sulla finestra tutti i sistemi di input (tastiera e mouse) vengono "catturati" da VirtualBox, per "liberarli" premere il tasto CTRL a destra della barra spaziatrice***

---

## Accedere velocemente

- È possibile predisporre l'ambiente per consentire un accesso più rapido e sicuro alle VM
  - Alias dei comandi ssh
  - Login con chiave pubblica
  - Configurazione di sudo

# Alias

- Per definire alias (comandi personalizzati) che evitino di digitare ogni volta il comando ssh con l'IP corretto, editare *sulla macchina host* il file `~/ .bashrc`
- Inserire le righe:  
`alias Client='ssh -X las@192.168.56.201'`  
`alias Router='ssh -X las@192.168.56.202'`  
`alias Server='ssh -X las@192.168.56.203'`

# Login

- Anzichè effettuare il login con password, si può utilizzare l'accesso ssh con chiave pubblica
- **Sulla macchina host** eseguire i seguenti comandi  
`ssh-keygen -t rsa -b 2048`
  - Premere invio ad ogni domanda
  - per Client (X=1), Router(X=2), Server (X=3) lanciare sull'host questo comando (a VM accese) :  
`scp .ssh/id_rsa.pub las@192.168.56.20X:`
- **per ogni VM**, collegarsi come *las* ed eseguire  
`mkdir .ssh`  
`chmod 700 .ssh`  
`cat id_rsa.pub >> .ssh/authorized_keys`

# Uso degli account

- 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
  - esistono modi diversi per operare come root all'occorrenza
    - *comando **su*** (switch user) permette di "diventare" l'utente specificato come parametro, a patto di conoscerne la password
    - *comando **sudo*** (super-user do) permette di eseguire un singolo comando come root, se abilitati, senza conoscerne la password
  - La seconda modalità è preferibile alla prima, però è necessario autorizzare in /etc/sudoers l'utente las

---

## sudo

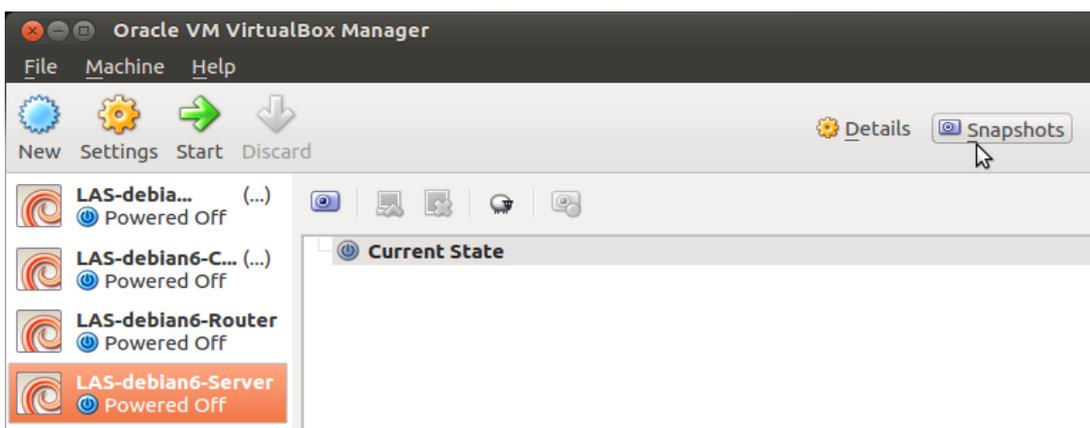
- Anzichè usare su per diventare root, configurare *sudo*
  - Diventare root con **su -**
  - Lanciare il comando **visudo**
- Deve essere presente una riga tra queste due:
  - **las ALL=(ALL:ALL) ALL**
    - dà accesso a sudo all'utente LAS
  - **%sudo ALL=(ALL:ALL) ALL**
    - dà accesso a sudo ai membri del gruppo sudo
    - in questo caso verificare che l'utente las sia nel gruppo sudo (comando id)
- Se non esiste, aggiungerla (preferibile la seconda variante, se esiste il gruppo e las ne è membro)

# Prima predisposizione (4)

- Le VM in questo modo nascono identiche, al primo avvio vanno differenziate editando tre file
    - `/etc/hostname` , `/etc/hosts`  
(deve contenere ovviamente il nome della VM: Client, Router, o Server al posto di base)
    - `/etc/network/interfaces`
      - sostituire **allow-hotplug eth0** con **auto eth0**
      - l'ultimo byte dell'IP di eth3 deve essere:
        - 201 per Client
        - 202 per Router
        - 203 per Server
- e cancellando (se esiste) il file  
`/etc/udev/rules.d/70-persistent-net.rules`
- Riavviare la VM

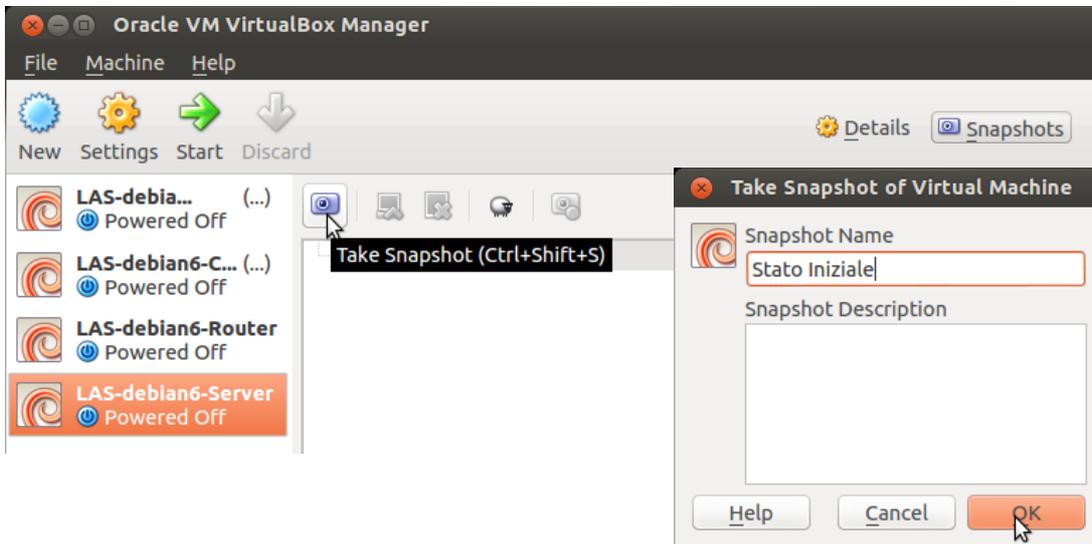
# Prima predisposizione (5)

- A questo punto le VM sono pronte.
- In caso di danno ad una VM, basta **eliminarla e ripetere la clonazione** (slide 3 e 4 della prima predisposizione)
- Se si vuole velocizzare il ripristino, basta creare uno **snapshot** dei 3 cloni dopo la configurazione specifica
  - Selezionare una VM e passare alla vista snapshot



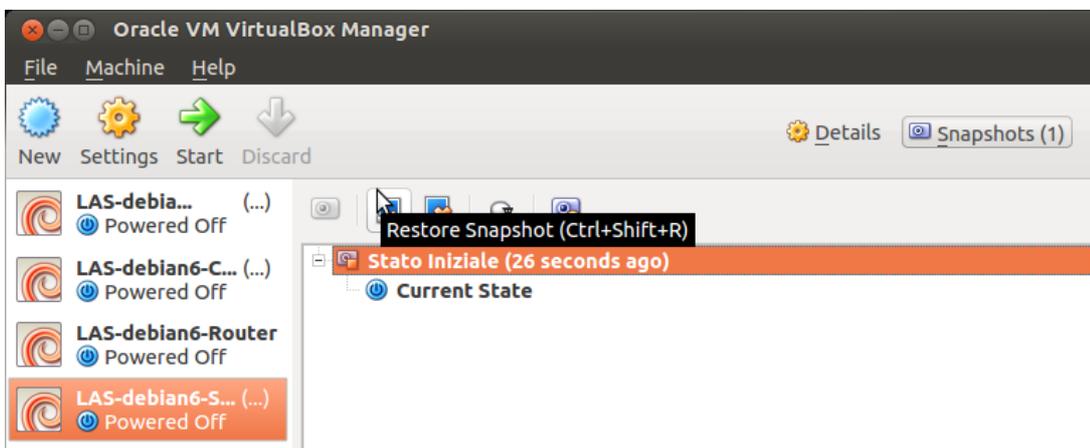
# Prima predisposizione (6)

- Con il tasto “fotocamera” scattare uno snapshot e chiamarlo, ad esempio, “Stato iniziale”



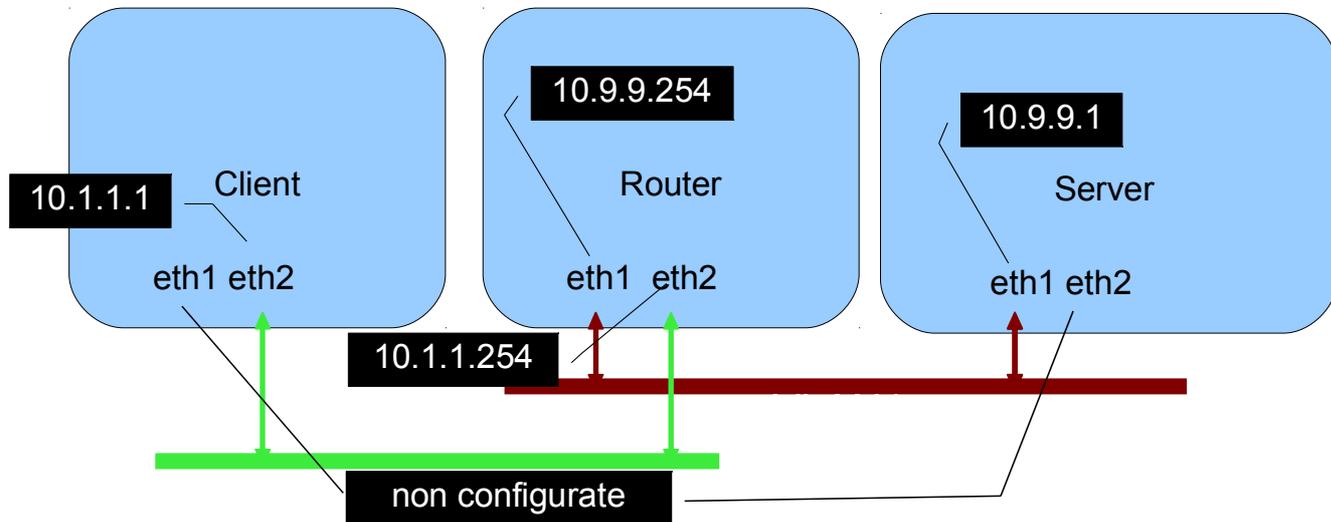
# Prima predisposizione (7)

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





# Esempio d'utilizzo



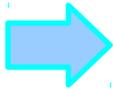
## Organizzazione

- È utile disporre di diversi terminali per ogni macchina
  - Dall'host è possibile contattare le macchine virtuali, per cui aprendo un terminale e digitando **ssh las@192.168.56....** si possono avere quanti terminali si vuole
- Alcune esercitazioni prevedono l'uso contemporaneo di diverse macchine virtuali (es. per svolgere in una rete i ruoli di client - router - server)
  - affollamento e confusione del desktop host
  - suggerimento: tenere le diverse macchine virtuali ed i loro rispettivi terminali su desktop virtuali diversi dell'host

# Gestione dei file

## Le macchine virtuali

- non hanno accesso a tutto ciò che è visibile dall'host, ad esempio ai dispositivi USB
- non dispongono di browser web (indispensabili in sede d'esame per la consegna dei file testati sulle MV)
- non dispongono di editor grafici, solo di *vi*



Indispensabile trasferire file da host a MV e viceversa

## Copia dei file (1)

- Via rete (più simile al caso reale)

```
scp file_su_host \  
    las@192.168.56.201:/path_su_guest/
```

```
scp \  
    las@192.168.56.201:/path/file_su_guest \  
    file_su_host
```

# Copia dei file (2)

- Shared folder

- La directory **/tmp/virtualbox** sull'host è mappata sul device virtuale **host** di tutti i guest
- ogni volta, prima di avviare VirtualBox creare **/tmp/virtualbox** sull'host
- Una volta per tutte
  - creare una directory **/mnt/host** sul guest
  - mount manuale:

**mount -t vboxsf host /mnt/host**

- mount automatico: inserire nel file **/etc/fstab**

**host /mnt/host vboxsf defaults 0 0**