

# Virtualizzazione e installazione Linux

Marco De Bona, Davide Quaglia

Lo scopo di questa esercitazione è quello di introdurre il concetto di virtualizzazione, di creare un ambiente virtuale con l'ausilio del software VirtualBox e di installare al suo interno il sistema operativo Linux.

## Cos'è la virtualizzazione?

Per virtualizzazione si intende la creazione di una versione virtuale di una risorsa normalmente fornita fisicamente. Qualunque risorsa hardware o software può essere virtualizzata: sistemi operativi, server, memoria, spazio disco, sottosistemi.

È quindi possibile eseguire, su una sola macchina (detta sistema host), altri sistemi operativi (detti sistemi guest) simultaneamente, ad esempio mettendo a disposizione un sistema Windows e un sistema Linux contemporaneamente.

## Perché virtualizzare?

Le motivazioni che spingono alla virtualizzazione di un sistema sono diverse, a seconda di chi si appresta a virtualizzare un sistema.

Gli sviluppatori hanno la possibilità di creare degli ambienti di prova nei quali far girare i loro software senza intaccare l'integrità del loro sistema, racchiudendo quindi i loro test all'interno di un'area controllata e priva di rischi.

Nell'ambito server è possibile avere server diversi, che offrono servizi diversi, sulla medesima macchina fisica, risparmiando sull'acquisto di nuove macchine server.

Vedremo come si utilizzano due programmi di virtualizzazione, per creare una macchina virtuale su cui installare sopra un sistema operativo Guest.

Nel nostro caso il sistema host è chiaramente la macchina sulla quale lavorate abitualmente, quindi basata sulla distribuzione Linux chiamata Ubuntu.

Sulla macchina guest installeremo Ubuntu 16.04.04 LTS da 64 bit ( Attualmente è disponibile anche la nuova versione 18.04 LTS) la quale **ISO** è scaricabile dal sito <https://www.ubuntu-it.org/download> .

*Immagine ISO .iso è una estensione utilizzata generalmente per file immagine, cioè un file che contiene l'intero contenuto di un archivio, che può essere usato direttamente (tramite software di emulazione) oppure scritto su un supporto ottico (CD o DVD ed esempio) tramite il processo di masterizzazione.*

Verranno presentati due programmi di virtualizzazione:

- Virtual-Manager, Open Source
- VirtualBox

Dal momento che si vuole dare un'impronta totalmente Open Source a questa esercitazione, se si è riusciti ad installare correttamente Virtual-Manager consigliamo di procedere con l'esercitazione utilizzando questo programma come virtualizzatore.

## Virtual Manager:

Il software di virtualizzazione che vedremo in questa esercitazione si chiama Virtual Manager (<http://virt-manager.org/>).

Si tratta di un front-end grafico per gestire diversi sistemi di virtualizzazione con una singola modalità grafica di interazione. I sistemi supportati sono:

- KVM/QEMU Linux hypervisor
- Xen hypervisor on Linux and Solaris hosts.
- LXC Linux container system
- OpenVZ Linux container system
- User Mode Linux paravirtualized kernel
- VirtualBox hypervisor
- VMware ESX and GSX hypervisors
- VMware Workstation and Player hypervisors
- Microsoft Hyper-V hypervisor
- IBM PowerVM hypervisor
- Parallels hypervisor
- Bhyve hypervisor
- Virtual networks using bridging, NAT, VEPA and VN-LINK.
- Storage on IDE/SCSI/USB disks, FibreChannel, LVM, iSCSI, NFS and filesystems

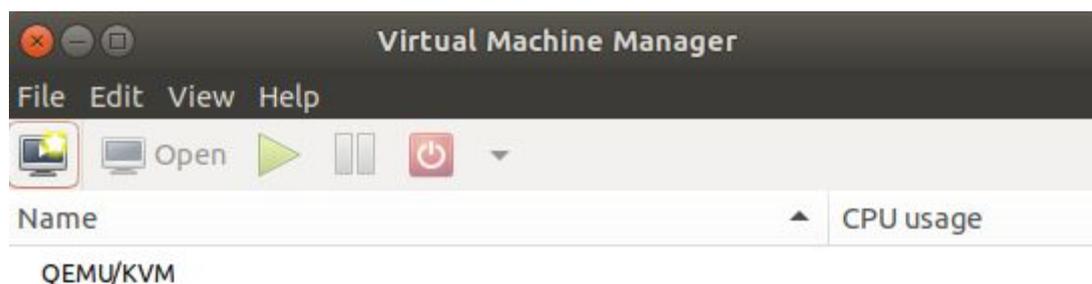
Virtual manager consente di creare nuove macchine virtuali e di monitorarne il funzionamento.

## Prima parte

### Creazione di una macchina virtuale con Virtual-Manager

Iniziamo con l'avviare Virtual Manager. Quello che dovrebbe comparirvi è una schermata come in figura1, dove QEMU/KVM sono gli Hypervisor della nostra virtual machine:

*Hypervisor* conosciuto anche come *virtual machine monitor*, è il componente centrale e più importante di un sistema basato sulle macchine virtuali, e svolge le attività di controllo dei sistemi operativi.



*figura1*

La schermata di gestione è veramente molto semplice.

Sulla sinistra c'è l'elenco delle macchine virtuali (adesso vuoto) e trovate poi l'icona per creare una nuova VM. 

Cliccate su tale icona e quello si apre è un Wizard per la creazione e configurazione della macchina virtuale.

## Step 1

Chiede come installare il sistema operativo. Scegliete come sorgente di installazione l'immagine ISO o il CDROM come in figura2.

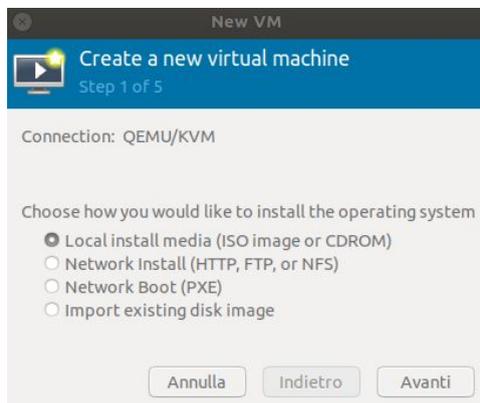


Figura2

## Step 2

La schermata successiva permette di scegliere se un CD è inserito nel lettore CDROM del sistema host oppure se un'immagine ISO è fornita nel file system del sistema host. Inoltre, rimuovendo la spunta dall'impostazione "Automatically detect operating system based on install media", è possibile specificare che tipo e versione di sistema operativo viene installato. Questa informazione non è strettamente indispensabile (secondo la definizione di VM qualsiasi sistema operativo deve essere eseguito) ma aiuta il virtualizzatore a compiere ottimizzazioni specifiche per il particolare sistema operativo.

Nel vostro caso lasciate che il sistema determini automaticamente il sistema operativo e prendiamo la ISO dalla cartella nella quale abbiamo precedentemente scaricato l'immagine ISO del nostro sistema Linux.

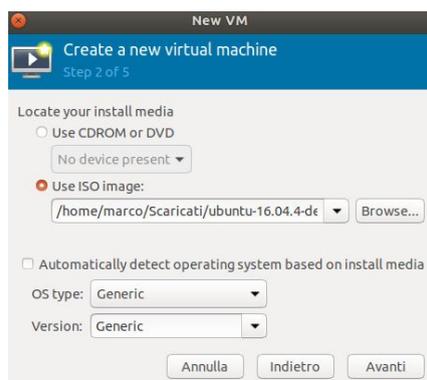


Figura3

### Step 3

Dobbiamo poi decidere l'ammontare di RAM che vogliamo che la macchina virtuale utilizzi.

Fate molta attenzione a scegliere il giusto quantitativo di RAM da dare alle vostre macchine virtuali; troppa RAM rischia di rallentare esageratamente il sistema host impedendovi di riprendere il controllo dell'intero sistema, troppa poca, invece, renderà il sistema guest non reattivo e in alcuni casi addirittura non avviabile.

Per le nostre esigenze impostate il quantitativo di RAM ad almeno 1024 MB e il numero di CPU utilizzate a 2 come in figura 4. ( Se le risorse del vostro computer sono limitate è comunque possibile impostare i parametri ad un valore minore )

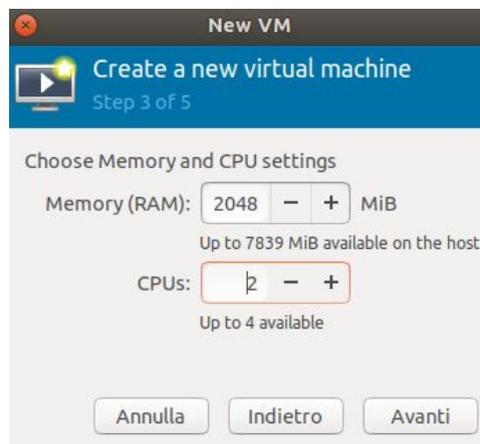


Figura4

### STEP 4

Dobbiamo ora configurare le impostazioni per l'Hard Disk che corrisponde ad un file che occupa memoria sul filesystem della macchina host.

Come prima cosa ci viene chiesto se vogliamo abilitare l'uso della memoria di massa per questa VM. Successivamente viene chiesto se creare un nuovo disco o se vogliamo usare un disco già esistente. Essendo il primo avvio è chiaro che non possiamo avere dischi esistenti da usare, passiamo quindi a crearne uno nuovo della dimensione di 15 GByte.

Virtual Manager chiede se allocare subito l'intero spazio per il disco. Ci sono infatti due approcci:

- ad espansione dinamica (se si toglie il segno di spunta)
- a dimensione fissa (se si lascia il segno di spunta)

Per l'**espansione dinamica** si intende un disco che inizialmente occupa dimensioni ridotte, e che cresce di volta in volta fino al raggiungimento delle dimensioni specificate.

A **dimensione fissa**, invece, si intende che il disco occupa subito tutto lo spazio che gli viene specificato in fase di creazione.

Entrambe queste scelte possono trovare un loro utilizzo specifico. I dischi ad espansione dinamica possono tipicamente risultare più lenti, perchè ogni volta che si riempie spazio sulla macchina virtuale, è necessario ampliare il file sul filesystem del computer host. Possono però tornare utili nell'ottimizzazione dello spazio su disco, infatti i dischi ad espansione dinamica occupano solo lo

spazio effettivo utilizzato dal sistema guest, lasciando libero eventuale spazio inutilizzato per altri scopi. I dischi ad espansione fissa, invece, risultano essere più veloci nelle operazioni di scrittura su disco e trovano quindi il loro impiego in quei settori in cui il millisecondo è essenziale per fornire un servizio (ad esempio nell'hosting di server).

Dal momento che l'allocazione dinamica provoca generalmente una lentezza maggiore, creiamo direttamente il nostro disco a dimensione fissa. Considerate che di default Ubuntu richiede più o meno 9 Gb di memoria solo per installare il sistema operativo, create quindi un disco di almeno 15Gb come in figura 5.



Figura5

## STEP 5

Andando avanti, compare un riepilogo delle impostazioni e la possibilità di cambiare le impostazioni dell'interfaccia di rete.

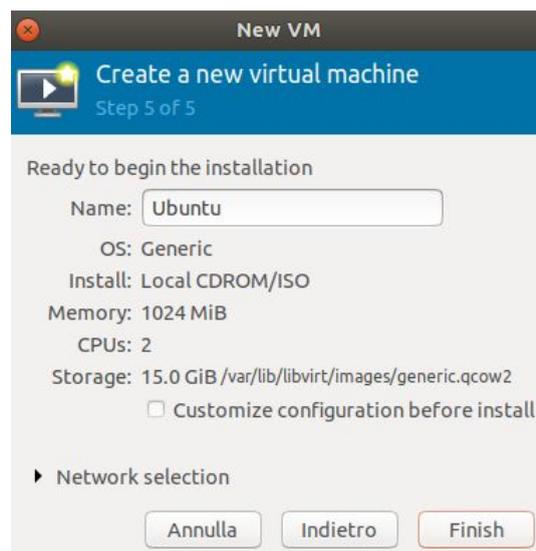


Figura6

La schermata di VirtualManager dovrebbe avere sulla sinistra la macchina appena creata, con il nome scelto, lo stato "spenta" e il grafico con l'utilizzo della CPU di tale macchina.

NOTA: In alcuni casi viene direttamente avviata la virtual machine.

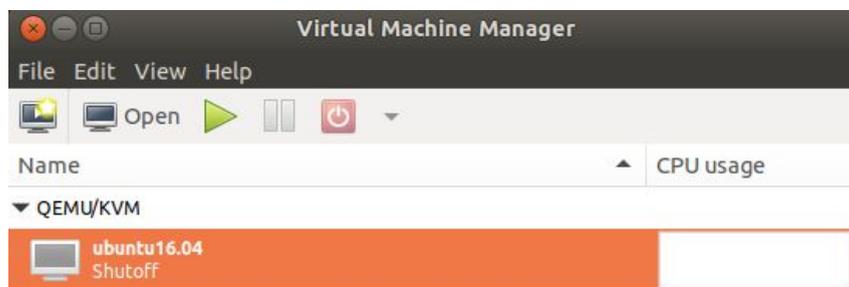


Figura7

A questo punto è come se avessimo a disposizione una macchina vera e propria, sulla quale però non c'è installato nessun sistema operativo. E' quindi necessario installarne uno per poter usare la macchina.

Di solito per installare un sistema operativo occorre inserire un CD nel lettore e avviare la macchina. Distribuire un CD a tutti gli studenti sarebbe stato un inutile dispendio di plastica in quanto con una macchina virtuale è possibile una soluzione più efficiente, ovvero quella di usare l'immagine ISO inserita virtualmente nel lettore CD della una macchina virtuale (Step 2).

## Seconda Parte

### Creazione di una macchina virtuale con VirtualBox

Avviata VirtualBox, se non avete ancora creato delle macchine virtuali dovreste trovarvi come in figura 8

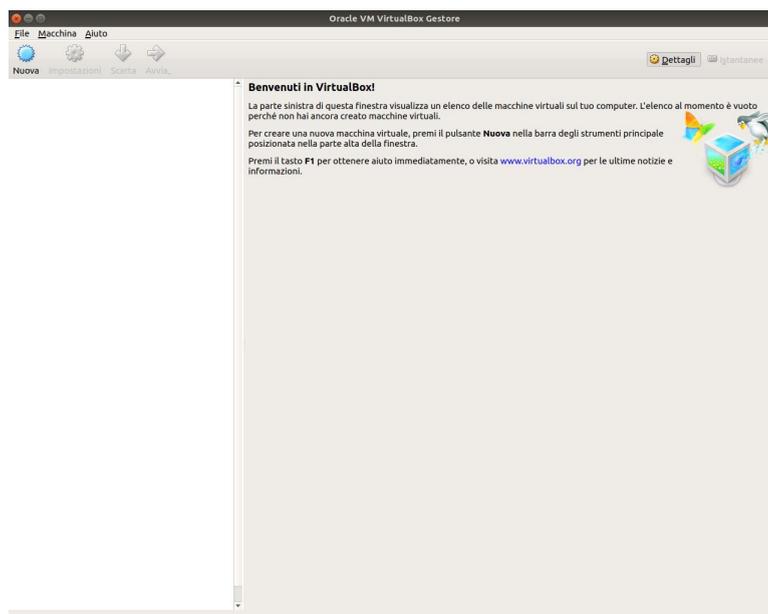


figura8

Per dare il via alla creazione della nostra macchina virtuale premete su “Nuova”.

## Step 1

Inizialmente viene chiesto con che nome chiamare la nostra macchina e il tipo di sistema operativo che stiamo per andare ad installare, procedete come in figura 9



Figura9

## Step 2

Ora, bisogna decidere l'ammontare di RAM che vogliamo che la macchina virtuale utilizzi. Fate molta attenzione a scegliere il giusto quantitativo di RAM da dare alle vostre macchine virtuali; troppa RAM rischia di rallentare esageratamente il sistema host impedendovi di riprendere il controllo dell'intero sistema, troppa poca, invece, renderà il sistema guest non reattivo e in alcuni casi addirittura non avviabile. Per le nostre esigenze impostate il quantitativo di RAM ad almeno 1024 MB.( Se le risorse del vostro computer sono limitate è comunque possibile impostare i parametri ad un valore minore )



Figura10

### Step 3

Viene chiesto ora di configurare le impostazioni per l'Hard Disk, che corrisponde ad un file che occupa memoria sul filesystem della macchina host.

Come prima cosa ci viene chiesto se vogliamo abilitare l'uso della memoria di massa per questa VM. Successivamente viene chiesto se creare un nuovo disco o se vogliamo usare un disco già esistente. Essendo il primo avvio è chiaro che non possiamo avere dischi esistenti da usare, passiamo quindi a crearne uno nuovo della dimensione di 15 GByte.

Virtual-Box chiede mette a disposizione 3 modalità per la creazione del disco, ma dal momento che non abbiamo un disco fisso virtuale già presente nel nostro computer, creiamo subito un nuovo disco fisso, lasciando VDI (passo successivo) come impostazione predefinita.



*figura 11*

Si ha ora la possibilità di scegliere tra due tipologie di modalità di archiviazione:

- **Espansione dinamica**, creazione di disco che inizialmente occupa dimensioni ridotte, e che cresce di volta in volta fino al raggiungimento delle dimensioni specificate.
- **Dimensione fissa**, disco occupa subito tutto lo spazio che gli viene specificato in fase di creazione.

Entrambe queste scelte possono trovare un loro utilizzo specifico. I dischi ad espansione dinamica possono tipicamente risultare più lenti, perchè ogni volta che si riempie spazio sulla macchina virtuale, è necessario ampliare il file sul filesystem del computer host. Possono però tornare utili nell'ottimizzazione dello spazio su disco, infatti i dischi ad espansione dinamica occupano solo lo spazio effettivo utilizzato dal sistema guest, lasciando libero eventuale spazio inutilizzato per altri scopi. I dischi ad espansione fissa, invece, risultano essere più veloci nelle operazioni di scrittura su disco e trovano quindi il loro impiego in quei settori in cui il millisecondo è essenziale per fornire un servizio (ad esempio nell'hosting di server).

Dal momento che l'allocazione dinamica provoca generalmente una lentezza maggiore, creiamo direttamente il nostro disco a dimensione fissa. Considerate che di default Ubuntu richiede più o meno 9 Gb di memoria solo per installare il sistema, create quindi un disco da almeno 15Gb

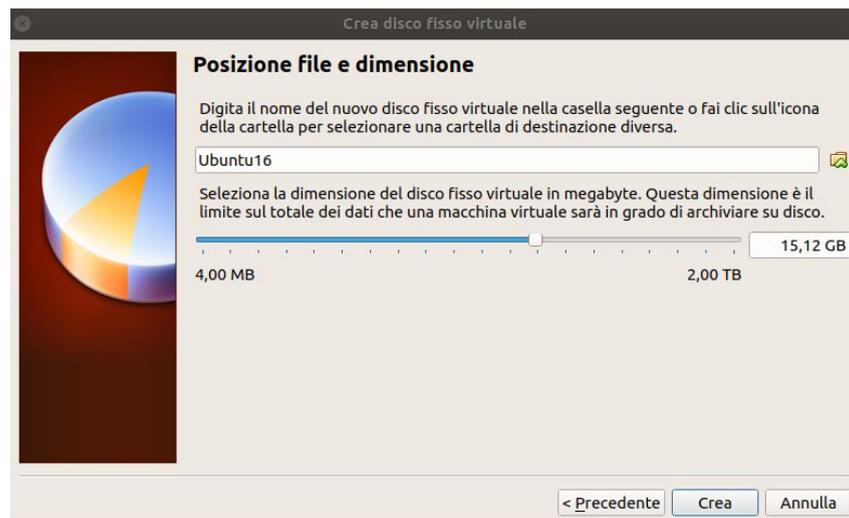


Figura11

A questo punto verremo rimandati alla schermata iniziale ma avremo a disposizione una macchina vera e propria, sulla quale però non c'è installato nessun sistema operativo. E' quindi necessario installarne uno per poter usare la macchina.

Di solito per installare un sistema operativo occorre inserire un CD nel lettore e avviare la macchina. Distribuire un CD a tutti gli studenti sarebbe stato un inutile dispendio di plastica in quanto con una macchina virtuale è possibile una soluzione più efficiente, ovvero quella di usare l'immagine ISO inserita virtualmente nel lettore CD della una macchina virtuale. Per inserire il disco ottico andare sul "reparto" archiviazione e cliccate sulla scritta "[Lettori ottici] Vuoto", dopodichè scegliete l'immagine ISO dalla locazione in cui l'avete precedentemente salvata. Se tutto è andato a buon fine dovrete essere in una situazione simile alla figura 12.

Avviate quindi la virtual machine.

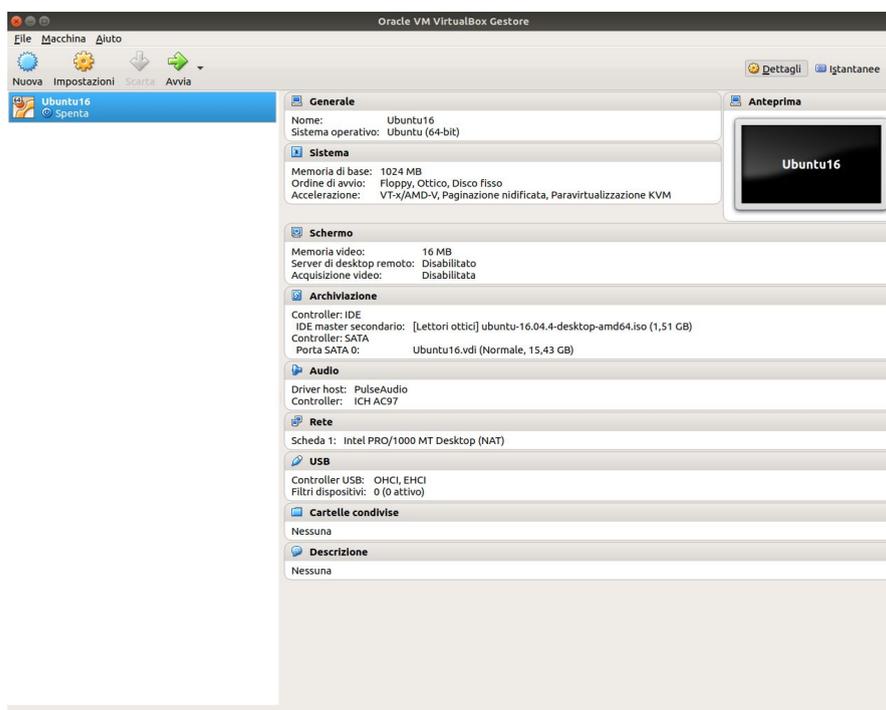


Figura12

## Seconda parte

# Installazione di Linux

Siccome il CD virtualmente inserito nella nuova macchina è un CD di boot allora viene eseguito un programma in esso contenuto che ha come scopo di installare il sistema operativo sulla macchina.

### STEP 1

All'avvio dovete selezionare la lingua in cui appariranno i messaggi e successivamente bisogna scegliere se installare sull'hard disk il sistema operativo o far partire quello presente sul CD (cioè la distribuzione cosiddetta "live").

L'avvio della modalità **Live** permette di visualizzare ed utilizzare il desktop di Ubuntu. Questa modalità è consigliabile quando si installa un sistema linux per la prima volta su un nuovo computer per accertarsi della piena compatibilità fra Ubuntu e i diversi componenti come ad esempio mouse, touchpad, scheda video, scheda di rete ecc. È comunque da tenere presente che i driver mancanti potrebbero essere comunque installati quando vengono scaricati i software di terze parti (vedi step 2) ed eventualmente è possibile verificare la presenza di eventuali driver proprietari da installare una volta terminata l'installazione di Ubuntu.

Per questa esercitazione possiamo partire direttamente con l'installazione del sistema, cliccate quindi sul relativo comando.

NOTA: Se state usando i computer del laboratorio la prima schermata di selezione lingua e modalità di avvio potrebbero apparire differenti. Procedere comunque seguendo le stesse istruzioni.

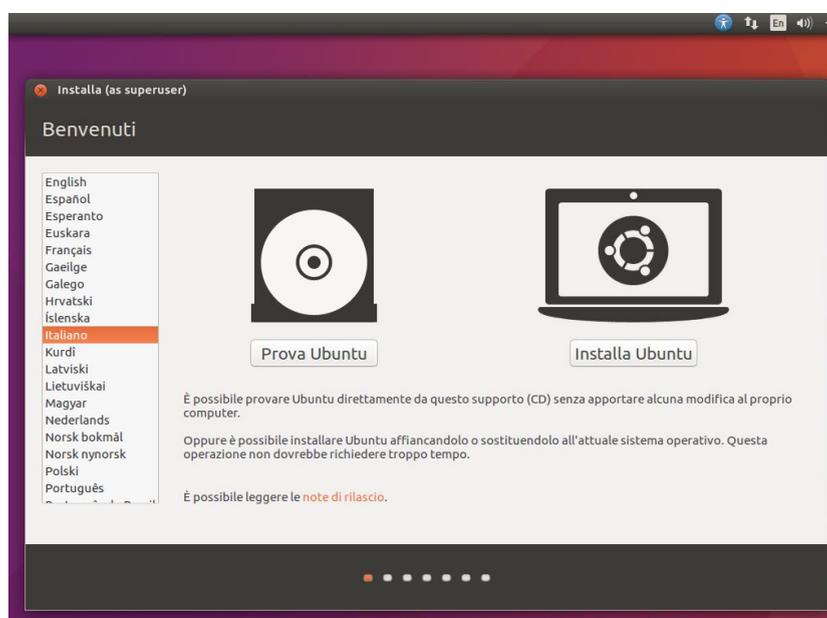


Figura13

### STEP 2

Selezionando l'opzione **Scaricare gli aggiornamenti durante l'installazione**, nel corso dell'installazione verranno installati anche aggiornamenti scaricati da Internet. Ciò può allungare sensibilmente il tempo necessario all'intero processo.

Selezionando l'opzione **Installare software di terze parti**, nel corso dell'installazione di Ubuntu verranno installati anche software non open source che consentono la riproduzione di contenuti

multimediali o il funzionamento di alcune periferiche.

Sia gli aggiornamenti che i software di terze parti possono essere installati ad installazione terminata. Consigliabile per quando si vuole installare il sistema più velocemente.

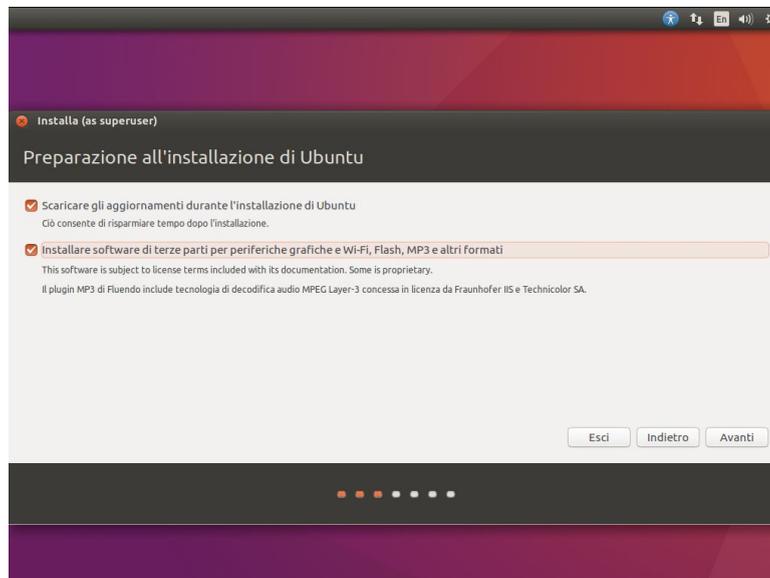


Figura14

### STEP 3

A questo punto si deve decidere come partizionare il disco.

Un disco fisico (hard disk, CD, DVD, memoria flash su SD Card o chiavetta USB) può essere diviso in massimo 4 **partizioni primarie** di cui una sola può essere una **partizione estesa** che può ulteriormente suddivisa in partizioni secondarie. Ogni partizione si comporta come un'unità fisica autonoma e può essere formattata in modo da ospitare un file system. Inoltre le partizioni primarie possono essere “bootable” cioè contenere la porzione di codice che viene eseguito all'avvio della macchina. Ogni dispositivo fisico contiene una **tabella delle partizioni** in esso contenute con le loro caratteristiche.

In un sistema operativo di tipo Unix (come Linux) il sistema delle cartelle di file è organizzato come un albero su cui vengono innestati (in gergo “montati”) gli alberi dei file system delle varie partizioni (di disco fisso, ma anche CD e chiavette USB inseriti nella macchina).

Un sistema basato su linux ha sostanzialmente bisogno di due partizioni distinte, una **primaria** sulla quale installare il sistema vero e proprio e l'altra detta **partizione di swap** che viene utilizzata come zona di memoria virtuale.

***Area di swap** Consente di estendere la capacità della memoria volatile complessiva del computer, oltre il limite imposto dalla quantità di RAM installata. L'uso dello swap è una delle tecniche impiegate dal sistema operativo per la gestione della memoria virtuale.*

Ubuntu, ci offre diverse soluzioni per partizionare il disco :

- Cancella il disco e installa Ubuntu, consente di eliminare tutti i dati presenti nel pc (sistemi operativi compresi ) ed installare Ubuntu come unico sistema operativo.
- Altro, consente di configurare manualmente le partizioni da creare, e modificare quelle eventualmente già esistenti, così da non cancellare i sistemi operativi già presenti sul computer.

Vediamo il partizionamento manuale; selezionate quindi **Altro** e andate avanti.

Comparirà la schermata di gestione delle partizioni che ci consente di aggiungere, modificare, eliminare o ripristinare le partizioni.

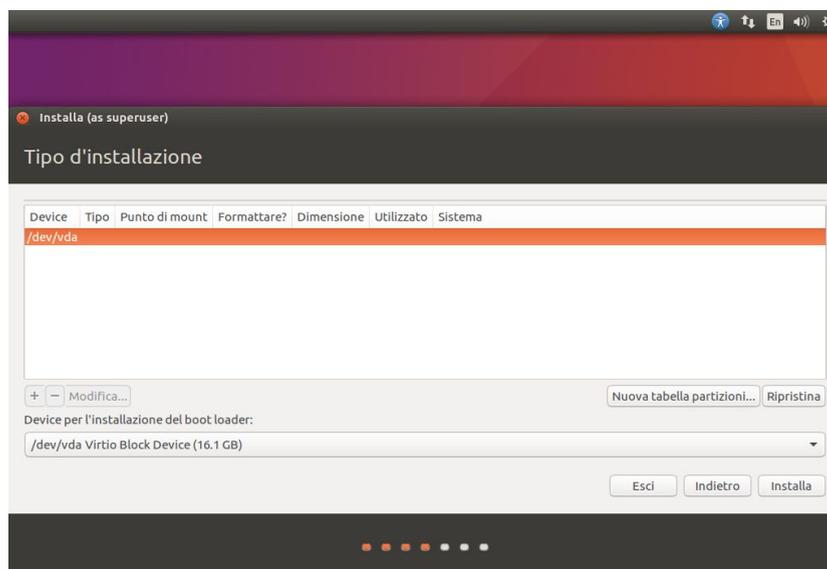


Figura15

Dalla schermata si vede che esiste un solo disco fisico, “/dev/vda”, e nessuna partizione (infatti l'hard disk è stato creato nuovo di zecca alla creazione della macchina virtuale). In Linux i dischi fissi, come tutti i componenti fisici, sono denominati come file della cartella /dev/.

**NOTA BENE:** il disco fisso /dev/vda non è il disco fisso della macchina host ma quello virtuale da 15 GB creato con Virtual Manager(o VirtualBox), infatti di seguito potete notare che la sua dimensione totale è proprio 15 GB.

Quello che vogliamo arrivare ad ottenere è un disco partizionato nel seguente modo:

- un sostanziosa parte del disco riservata ad Ubuntu.
- una parte più ridotta per la partizione di swap cioè l'area di disco vista dal sistema operativo come “memoria virtuale” cioè memoria RAM addizionale a quella realmente presente nella macchina.

Iniziamo quindi a selezionando il tasto **Nuova tabella partizioni** che consente di creare una nuova tabella che mantiene le informazioni relative alle partizioni del disco.

Comparirà una voce sotto a /dev/vda che identifica la presenza di uno spazio libero dal quale ricavare le partizioni. Selezionate lo spazio libero e poi su **Aggiungi**.

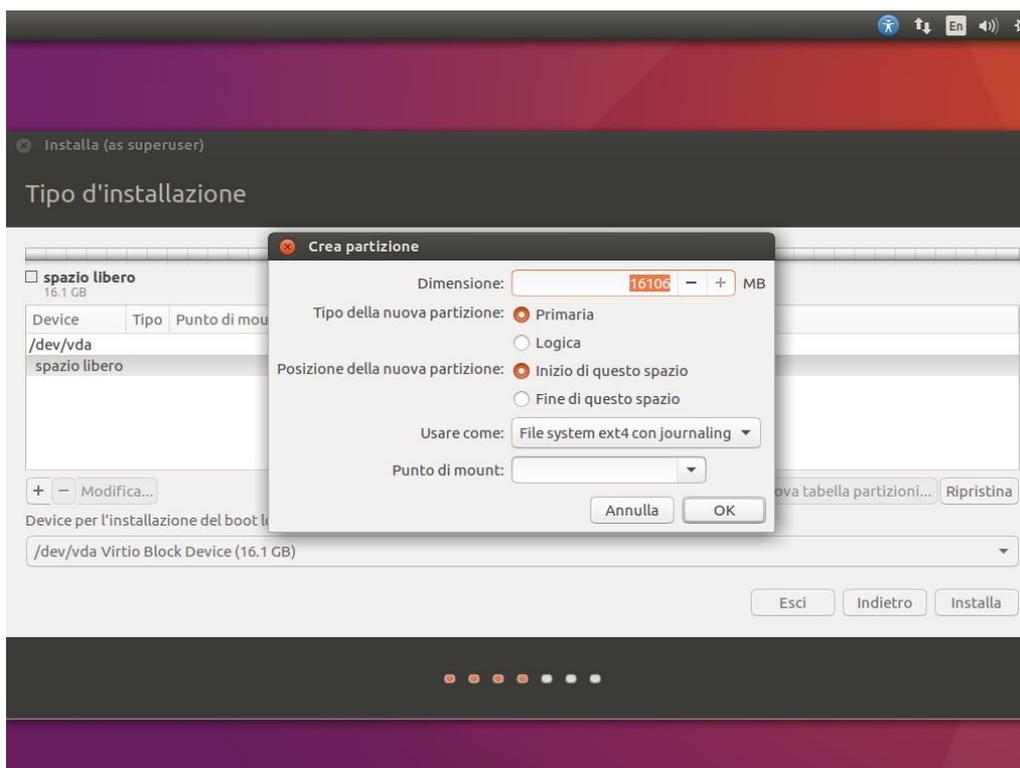


Figura16

Dobbiamo ora specificare le impostazioni relative alla prima partizione, quella sulla quale installeremo Ubuntu.

**Tipo della partizione:** Distinguiamo subito due tipologie di partizioni:

- Primaria (in cui risiede il sistema operativo, e' l'unico tipo di partizione "avviabile")
- Logica (è la partizione estesa che funge da contenitore per le partizioni "logiche", può essere visto come una specie di disco fisso che a sua volta deve essere partizionato);

Lasciamo quindi selezionata come tipologia “Primaria”.

**Dimensione partizione:** Settiamo ora la dimensione del disco, impostate 14000 come dimensione e lasciamo il resto per la partizione di swap (generalmente si usa dare almeno tanta Swap quanta è la memoria RAM del nostro sistema).

**Posizione della partizione:** dove deve essere posizionata la partizione in riferimento alla prima sezione fisica del disco (lasciate “inizio”)

**Usare come:** scelta del tipo di filesystem con il quale formattare il disco. Solitamente il filesystem utilizzato da linux si chiama “Extended file system con journaling” abbreviato in “ext” e attualmente siamo alla versione 4. Lasciate allora selezionato ext4 come tipo di filesystem.

**Punto di mount:** il punto di mount specifica il percorso logico sul quale viene montato il disco in fase di boot del sistema. Specifichiamo quindi come punto di mount root (“/”) ovvero il disco viene montato come radice del sistema.

Premete su OK per confermare, ecco cosa dovrete vedere:

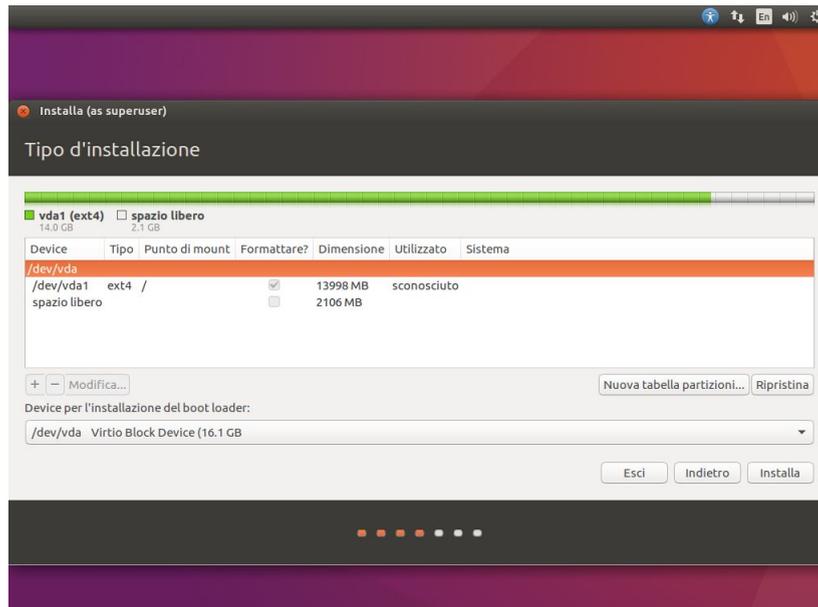


Figura17

Vedete che rimangono circa 2106 MB di spazio inutilizzato; procediamo come prima per creare una partizione con tale spazio rimasto per lo spazio di swap . Questa volta, però, dovete usare queste impostazioni:

- Tipo di partizione: Logica;
- Dimensione: viene automaticamente preso l'intero spazio libero;
- Posizione: fine;
- Usare come: scegliere “area di swap”;
- Punto di mount: non lo potete specificare perchè le zone di memoria riservate alla memoria virtuale non sono utilizzabili direttamente dall'utente;

confermate le modifiche ed ecco cosa dovrete vedere:

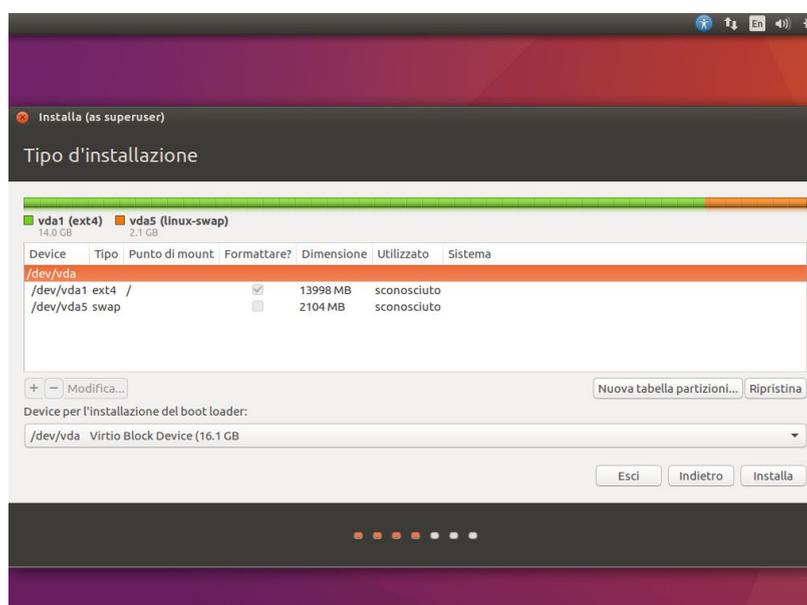


Figura18

Il gestore delle partizioni mostra un riepilogo delle modifiche specificate, una partizione primaria da circa 14000 MB e una da 2100 MB come partizione di swap.

Premete ora su **Installa**, vi verrà mostrato un riepilogo delle modifiche apportate, premete quindi **Avanti**. Successivamente vi verranno chieste località e impostazioni della tastiera. Andare semplicemente avanti.

NOTA: Nel caso steste installando un sistema linux in un computer nella quale è già presente un sistema operativo ( per esempio Windows 10 ), la vi comparirà una schermata come in figura 19. L'opzione **Installa Ubuntu a fianco di Windows** consente di ridimensionare la partizione di Windows per ricavare lo spazio necessario all'installazione di Ubuntu.

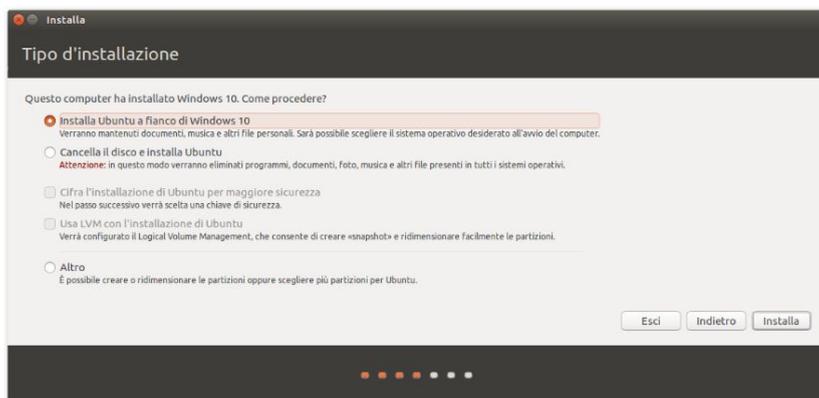


Figura19

#### STEP 4

Specificate i vostri dati personali quali:

- nome utente per il login,
- la password;
- il nome del computer;
- tipo di accesso. È possibile cifrare la cartella personale per una maggiore sicurezza dei dati. Tuttavia questa opzione diminuisce leggermente le performance del sistema e renderà più difficile l'eventuale recupero di file personali perduti accidentalmente. Per questa esercitazione inserire una password ma non cifrare la cartella personale.

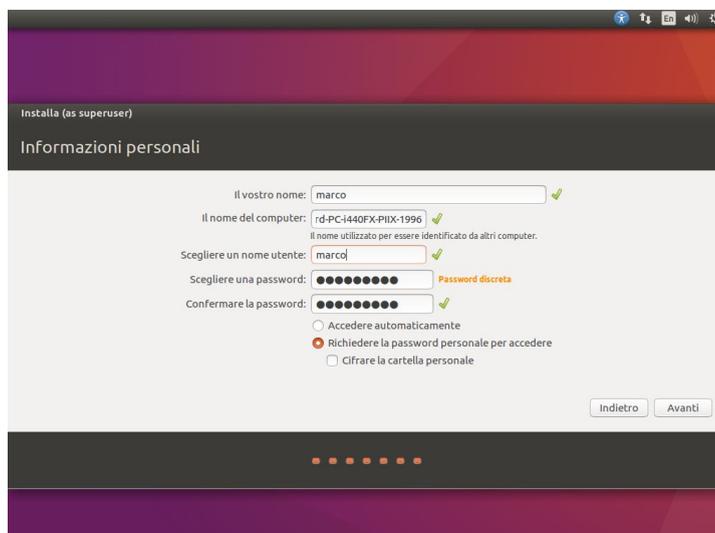


Figura20

Andando avanti l'installazione del sistema operativo avrà inizio. Al termine di tale installazione verrà chiesto di riavviare la macchina e in fase di riavvio viene chiesto di rimuovere il disco e premere invio.

NOTA BENE: non bisogna riavviare la macchina host del laboratorio ma la macchina virtuale creata con Virtual Manager. Il programma di installazione si occupa di spegnere e riaccendere la macchina virtuale come se fosse una macchina reale. Inoltre il boot da disco fisso (virtuale) ha la priorità su quello da CD e per questo motivo partirà il sistema operativo appena installato.

Una volta effettuato il login avete a disposizione un sistema separato da quello host sul quale fare esperimenti test e quanto più vi pare.

L'installazione potrebbe richiedere un pò di tempo. Si consiglia di andare all'Appendice A per un ripasso del **file system Linux** prima di cominciare con la terza parte.

NOTA: Se avete utilizzato VirtualBox non appena è finita l'installazione e siete entrati col vostro account cliccate sulla finestra "Dispositivi" ed inserite **L'immagine del CD delle Guest Additions**.

## Terza parte

# Introduzione ai server Linux - APACHE2

### 1 Linux Server

Dal punto di vista dei sistemi operativi desktop, ovvero quelli utilizzati giornalmente dagli utenti, Windows è tutt'oggi quello che raccoglie il maggior numero di seguaci, ma in ambito server le cose sono ben diverse. Già da qualche anno infatti nell'**ambiente server**, è Linux a fare da padrone, lasciando gli inseguitori ad una notevole distanza.



Figura20

Attualmente la tendenza delle aziende è sempre quella di adottare un sistema server basato su Linux grazie a 4 caratteristiche dei loro server:

- **Stabilità:** ricordiamo che una cosa molto importante per un server che offre un servizio è quello di non andare mai offline, avere quindi un sistema operativo nella quale il rischio di crash è ridotto non può che essere un vantaggio. Oltre ad una minor assenza di crash, Windows server spesso richiede un riavvio dopo gli aggiornamenti, cosa che su Linux invece non avviene.
- **Sicurezza:** L'architettura del sistema lo rende più sicuro per natura, oltre al fatto che attacchi di virus e malware sono molto meno frequenti.
- **Hardware:** Generalmente le risorse impegnate dai sistemi Linux sono inferiori a quelle richieste dagli altri sistemi operativi.
- **Costi:** Open source!!

## 2 Web Server

Un **web server** è un'applicazione software che, in esecuzione su un server, gestisce i trasferimenti delle pagine web ai client (browser) che ne fanno richiesta.

Il protocollo con la quale la maggior parte dei server web lavora è l'**HTTP (Hyper Text Transfer Protocol)**, in quanto supportato dalla stragrande maggioranza dei client (PC, smartphone, tablet ecc.). Un web server generalmente attende connessioni TCP mettendo in ascolto utilizzando un Socket associato a una "porta ben nota" (in genere ma non necessariamente la porta 80 per i server HTTP). Quando un client si connette, il server riceve una notifica (in modi dipendenti dall'OS) e può quindi accettare il socket in ascolto, creando così un nuovo socket.

Una volta la connessione tra client e server è stata instaurata e la nostra **richiesta http** è stata consegnata al server indicato, quest'ultimo ci manderà una **risposta http** e un file HTML che verrà tradotto dal browser del client sotto forma di testo e immagini.

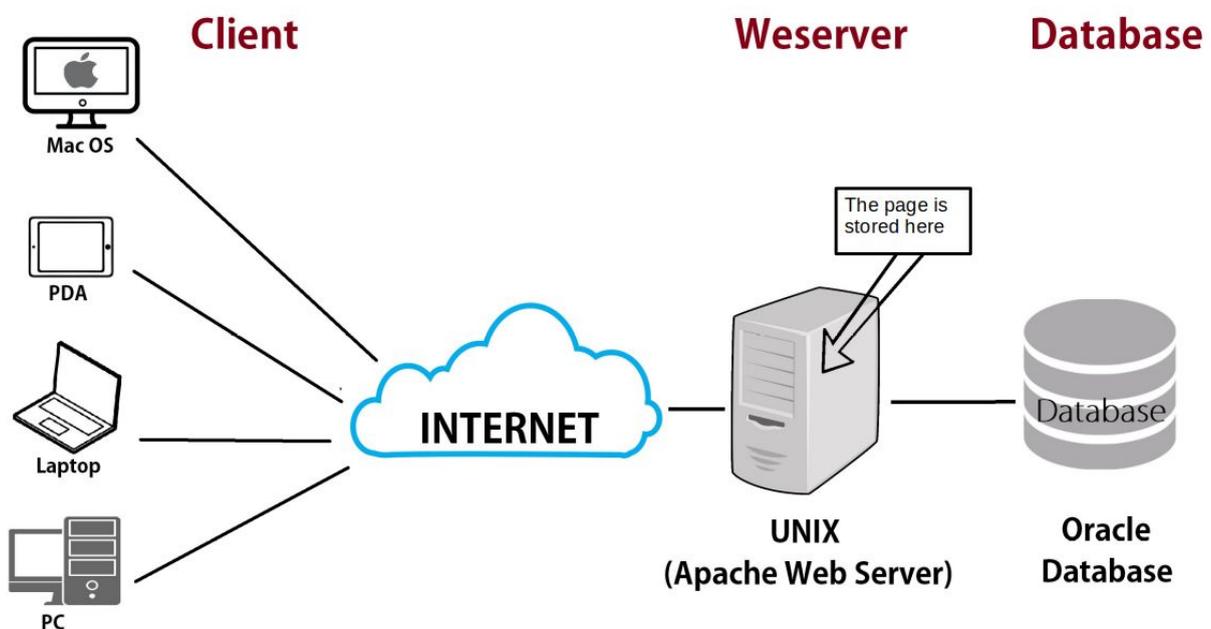


Figura21

Le pagine web interrogate si suddividono in due categorie, **statiche** e **dinamiche**.

- Le pagine statiche contengono solamente codice HTML che al suo interno contiene tutto quello che verrà mostrato all'utente finale.
- Le pagine dinamiche, invece, oltre ad HTML usano altri linguaggi (PHP, ASP, ecc..) attraverso i quali sono definite delle istruzioni dove vengono generati i contenuti della pagina richiesta.

Per proseguire con la filosofia Open source, il software che andremo ad utilizzare, per gestire quanto visto fino ad ora, è **APACHE2**, ovvero il web server più utilizzato al mondo.

Per motivi di tempo in questa esercitazione noi vedremo e utilizzeremo solamente pagine statiche.

NOTA : Esistono tanti tipi di configurazione di un web server. La configurazione standard prende il nome di **LAMP**

- **Linux**: sistema operativo host.
- **Apache2**: software che fa da web server.
- **MySQL**: sistema per la gestione della basi di dati.
- **PHP**: Linguaggio per la creazione delle pagine dinamiche

### 3 APACHE2

#### 3.1 Installazione

Per installare APACHE2 è necessario digitare un semplice comando da terminale :

```
$ sudo apt-get install apache2 apache2-doc
```

Il primo pacchetto è quello necessario per il funzionamento, mentre il secondo è facoltativo ma consigliato e consente di avere tutta la documentazione ufficiale (purtroppo non in italiano) all'interno del nostro server sempre pronta per essere consultata.

Una volta terminata l'installazione è possibile testare che tutto sia andato a buon fine collegandosi da un PC connesso alla stessa rete, al nostro server.

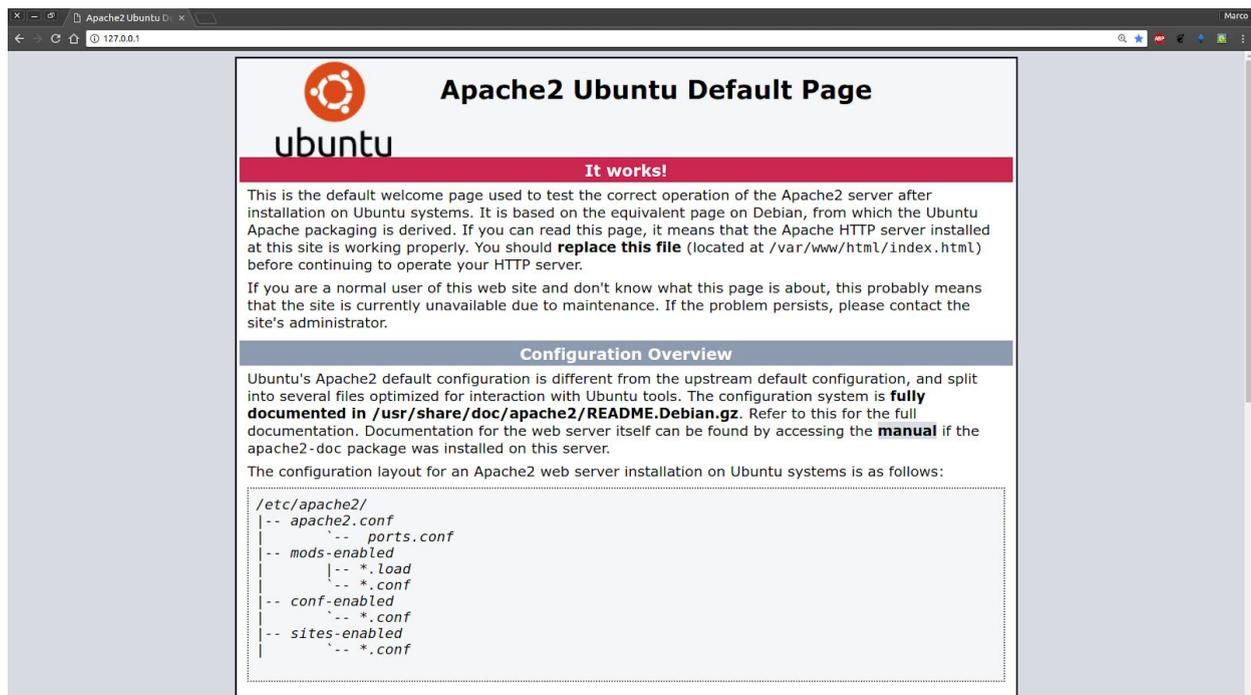


Figura22

#### Esercizio 1

Nell'immagine potete vedere che l'indirizzo inserito è il **localhost** in quanto si sta usando il browser

sulla stessa macchina che ospita il web server.

- Se avete saltato la parte dell'esercitazione sulla creazione del sistema Linux in quanto avete già un sistema Linux nel vostro computer, provate ad inserire nel vostro browser l'indirizzo IP ( ifconfig ) di un vostro compagno, se anche lui ha Apache2 installato ed in esecuzione, dovrebbe comparirvi di nuovo la pagina di default.
- Se invece state usando il sistema Linux virtualizzato, provate a fare la stessa cosa inserendo l'indirizzo IP del sistema guest nel browser del sistema host (bisogna aver configurato la VM con rete in modalità *bridge* che però con la rete UNIVR non funziona).

NOTA : Se avete installato anche il manuale provate ad inserire nel vostro browser [http://IP\\_DEL\\_SERVER/manual/](http://IP_DEL_SERVER/manual/) .

```
$ sudo /etc/init.d/apache2 start    #avvia il server
$ sudo /etc/init.d/apache2 stop    #stoppa il server
$ sudo /etc/init.d/apache2 restart #riavvia il server
```

## 3.2 Configurazione

La **configurazione di un server Apache2** viene gestita tramite direttive scritte su file di testo. In Ubuntu questi file e cartelle di configurazione li troviamo tutti all'interno della directory */etc/apache2*.

Noi vedremo solo alcuni dei file di configurazione, nell'Appendice A vengono riportati tutti i file di configurazione e le direttive che è possibile assegnargli.

In particolare i seguenti file e cartelle di configurazione:

- **apache2.conf**: è il file principale di configurazione. Al suo interno troviamo tutte le impostazioni *globali* ovvero quelle che saranno applicate a ciascun sito presente sul nostro server.
- **sites-available**: questa cartella racchiude i file di configurazione di tutti i Virtual Host (VH), ovvero tutti i siti presenti sul server. I VH consentono di avere più siti all'interno dello stesso web server come vedremo più avanti.
- **sites-enabled**: questa cartella contiene collegamenti simbolici a *sites-available* e rappresenta la lista dei siti effettivamente accessibili ai client.

Useremo le seguenti direttive:

- **ServerRoot**: definisce la directory principale per le configurazioni di Apache. Il valore di default è */etc/apache2*.

- **DocumentRoot**: indica la directory contenente il sito web richiesto al server. Nella nostra distro Ubuntu il valore predefinito è `/var/www`.
- **ServerAdmin**: indica l'indirizzo e-mail dell'amministratore del server. Comparirà nei messaggi di errore che i client ricevono.
- **ServerName**: indica l'hostname al quale il server Apache deve rispondere.

Dopo ogni modifica ad un file di configurazione è necessario riavviare il servizio per rendere effettive le modifiche.

In questo momento nel web server che sta girando sul nostro computer è già presente una pagina web, ovvero quella che ci avvisa della corretta installazione di Apache2.

Come indicato dalla direttiva Document Root, presente all'interno del file di configurazione `/etc/apache2/sites-available/000-default.conf`, il percorso fisico nella quale il nostro web server va a cercare la pagine web è `/var/www/html`. Se provate a raggiungere quest'ultimo percorso vi troverete al suo interno un file chiamato `index.html` che al suo interno contiene il codice HTML che genera la pagina di default vista in precedenza .

## Esercizio 2

Se prima quando avete provato ad inserire l'indirizzo IP del vostro compagno non siete riusciti ad apprezzare fino in fondo quello che era avvenuto, provate ora a ripetere la stessa azione dopo aver modificato il contenuto del file `index.html` ed averlo sostituito con dell'altro codice HTML.

Potenzialmente, se vogliamo che il nostro web server ospiti un solo sito web abbiamo praticamente tutto quello che ci interessa sapere, ma cosa succede se abbiamo bisogno di gestire più siti attraverso lo stesso web server?

## 3.3 Virtual Hosting

Abbiamo visto come configurare un sito sul nostro web server Ubuntu. Capita spesso, però, di imbattersi nella necessità di avere più siti web sullo stesso server, pur avendo disponibile un solo indirizzo IP pubblico.

Per far fronte a questa necessità, ci viene in aiuto quello che viene definito **Virtual Hosting**, ovvero la possibilità di eseguire più siti web su di un unico server. Gli host virtuali possono essere:

- *IP-based*: abbiamo un indirizzo IP per ciascun sito web;
- *name-based*: abbiamo più nomi che rimandano allo stesso indirizzo IP.

In entrambi i casi l'intera struttura è completamente trasparente per i client: il fatto che più siti girino sulla stessa macchina fisica non è visibile dall'utente finale.

Noi vedremo quella che è la configurazione tipica dei web server, ovvero quella name-based. Quello che andremo a fare è dire al nostro server Apache di distinguere il sito web da restituire al client in base allo specifico hostname interrogato.



Figura23

I passaggi da compiere sono i seguenti :

- Creare la cartella che ospiterà il nostro nuovo sito.
- Aggiungere alla directory sites-available un file di configurazione per il nuovo sito.
- Configurarlo ed abilitarlo.

```
$ sudo mkdir /var/www/test.com
```

```
$ sudo nano /etc/apache2/site-available/test.com.conf
```

All'interno di questo file che viene creato (se apriamo con **nano** un file che non esiste quest'ultimo verrà creato) andiamo ad inserire le direttive principali per il funzionamento del sito:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName www.test.com

    DocumentRoot /var/www/test.com

</VirtualHost>
```

Per abilitarlo infine è necessario creare un collegamento simbolico nella cartella di configurazione sites-enabled e per farlo basta un semplice comando:

```
$ sudo a2ensite test.com
```

Come sempre quando vengono apportate delle modifiche ai file di configurazione è necessario riavviare il server.

NOTA: `$ sudo a2dissite sitoX` permette di disabilitare il sito corrispondente.

Questa operazione può essere ripetuta per ogni sito che vogliamo aggiungere al nostro server web.

Naturalmente, prima che il sito sia effettivamente visibile sul web bisogna fare un lavoro che esula

dal contenuto di questa esercitazione. Vengono qui riassunti brevemente i passi che sarebbero da fare :

- Comprare un dominio
- Configurare i DNS aggiungendo i record necessari affinché il nostro dominio diventi raggiungibile
- Aprire le porte usate dal sito web tramite il firewall che gestisce il traffico della rete del nostro server

Per verificare il corretto funzionamento dei siti web che abbiamo inserito ci limiteremo a modificare il contenuto del file `/etc/hosts` inserendo al suo interno i siti creati:

```
127.0.0.1    localhost
127.0.1.1    marco-Standard-PC
127.0.0.1    www.test.com
127.0.0.1    www.test2.com
ecc..
# The following lines are desirable for IPv6 capable hosts
::1    ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Una volta chiuso il terminale non ci resta che andare sul browser e digitare localhost.

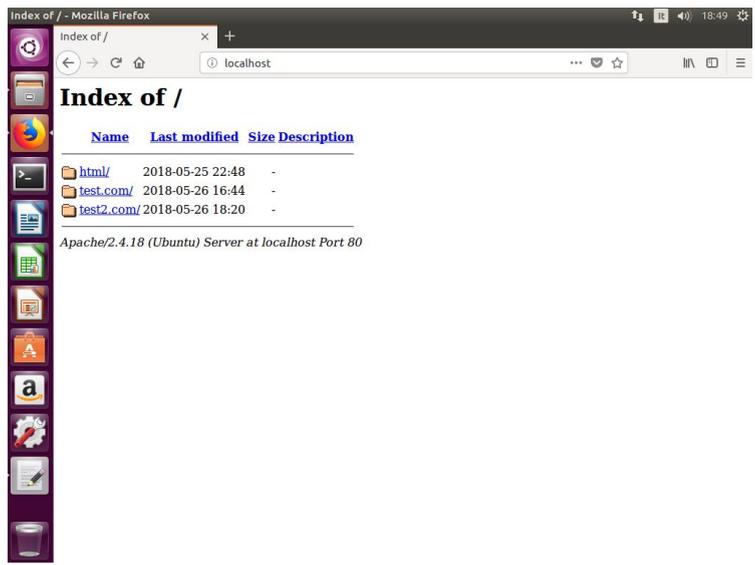


Figura24

## Appendice A - File System Linux

Il modo in cui Linux organizza i suoi file nel disco rigido, è molto diverso dal modo in cui Windows esegue lo stesso compito. I nuovi utenti di Linux, provenienti da Windows, a volte hanno difficoltà a gestire le directory o giungono alla convinzione che Linux gestisca i propri file nello stesso modo in cui vengono gestiti da Windows.

### **Tutto comincia con Root: /**

La directory root conosciuta semplicemente come "/" è il punto di partenza. La directory root agisce come il "C: Drive" in Windows. Un sistema Linux non può avviarsi senza una partizione root.

### **Tutto è in: /bin**

Contiene i programmi basilari per la gestione del sistema, in pratica buona parte dei comandi base utilizzabili dalla riga di comando da qualsiasi utente senza dover utilizzare i privilegi dell'amministratore.

### **Dove tutto comincia: /boot**

Contiene le immagini del kernel e i file indispensabili al bootstrap del sistema. In particolare il file /boot/grub/grub.cfg è fondamentale per l'impostazione dei parametri di avvio di **Grub** il bootloader.

### **Ogni dispositivo è un file: /dev**

È la directory che individua sotto forma di file le periferiche hardware.

Per esempio, un hard disk può essere qui individuato dal file /dev/sda e le sue partizioni attraverso i file /dev/sda1, /dev/sda2, ecc. Occorre tuttavia ricordare che in questa directory i dispositivi di memorizzazione non sono direttamente esplorabili. Per visionarne i contenuti si possono sfruttare le cartelle che Ubuntu crea automaticamente nella directory /media, o alternativamente effettuare il montaggio manualmente.

### **Configurazione: /etc**

Contiene i file di configurazione del sistema.

Fra i numerosi file è opportuno ricordare /etc/fstab, fondamentale per definire le partizioni da rendere disponibili all'avvio del sistema.

Segue un breve elenco con il contenuto di alcune delle sotto directory principali:

- /etc/apt: file di configurazione dei repository, in particolare /etc/apt/sources.list.
- /etc/grub.d: file per personalizzare la sequenza di boot del sistema.
- /etc/samba: /etc/samba/smb.conf file di configurazione per la condivisione dati tramite Samba.
- /etc/X11: file per la configurazione delle schede video.

### **Le librerie condivise: /lib**

Contiene tutte le librerie condivise del sistema.

### **Quando non si è spento correttamente il pc: /lost+found**

Contiene le parti di file recuperate al riavvio in caso di interruzione della corrente elettrica, nel caso in cui vengano persi i dati sui quali si stava lavorando

## **Dove vivono i "Mount Point": /mnt e /media**

/media

Secondo gli standard dovrebbe essere la directory destinata al montaggio dei dispositivi rimovibili. Tuttavia in Ubuntu viene utilizzata indistintamente per il montaggio automatico di qualsiasi dispositivo di memorizzazione (CD, DVD, chiavette USB, partizioni degli hard disk, ecc..).

Ogni volta che un dispositivo viene collegato al computer o dal file manager si seleziona una partizione dell'hard disk, il sistema crea all'interno della directory /media una cartella dalla quale è possibile esplorarne il contenuto.

/mnt

È secondo gli standard la directory destinata al montaggio dei dispositivi. Tuttavia in Ubuntu gli automatismi di sistema non la utilizzano a vantaggio della directory /media.

Continua tuttavia ad essere frequentemente utilizzata dagli utenti per i montaggi manuali.

## **La directory "Optional": /opt**

È la directory deputata a contenere programmi ottenuti da installazioni non standard. In realtà in Ubuntu non viene utilizzata, viene talvolta utilizzata dal singolo utente per l'installazione manuale di software non contenuti nel sistema dei repository.

## **Nella directory del kernel: /proc e /sys**

/proc

Contiene le informazioni relative al sistema, come un file system virtuale creato dal kernel dinamicamente istante per istante, in memoria e non sul disco

/sys

Similmente alla directory /proc funziona da file system virtuale. Al suo interno vengono visualizzati i dispositivi collegati e i moduli del kernel attivi.

## **Siete qui ed ora non più: /tmp**

Contiene file temporanei.

## **Dove vivono i Programmi: /usr**

È la directory che contiene la maggior parte dei programmi installati sul sistema.

Segue un breve elenco con il contenuto di alcune delle sotto directory principali:

- **/usr/bin**: file eseguibili delle applicazioni accessibili a tutti gli utenti, vale a dire i programmi normalmente avviabili dal menù delle applicazioni.
- **/usr/sbin**: file eseguibili delle applicazioni di sistema accessibili solo all'amministratore.
- **/usr/share**: file di vario genere (configurazione, documenti di testo, ecc..) indipendenti dall'architettura del sistema (i386, amd64). Ad esempio in /usr/share/nautilus/ui sono presenti file di configurazione dell'interfaccia di **Nautilus**; le cartelle backgrounds, icons e themes contengono sfondi, icone e temi del desktop; dal file /usr/share/synaptic/html/index.html si accede ad una guida in formato html su **Synaptic**, e così via...

- **/usr/src**: codice sorgente, ad esempio del kernel e dei relativi headers.

### **Non solo file temporanei: /var**

Contiene tutti file che hanno informazioni dinamiche, che tendono a modificarsi con il tempo: log, file di pid e lock dei processi in esecuzione, directory di spool (stampa, mail...) ecc.

Segue un breve elenco con il contenuto di alcune delle sotto directory principali:

- **/var/log**: file di registro sui quali viene registrato ciò che avviene al sistema in ogni istante. Consultare i file di log può aiutare a risalire alle cause di eventuali problemi nel sistema.
- **/var/cache/apt/archives**: durante il processo di installazione dei programmi è in questa cartella che viene memorizzata una copia di sicurezza dei pacchetti .deb scaricati. Dal momento che con il tempo possono arrivare ad occupare molto spazio di memoria, l'amministratore può prendere in considerazione la rimozione dei pacchetti, sia manualmente sia attraverso apt-get.

## **Appendice B**

File e cartelle di configurazione:

- **apache2.conf**: è il file principale di configurazione. Al suo interno troviamo tutte le impostazioni *globali* ovvero quelle che saranno applicate a ciascun sito presente sul nostro server.
- **ports.conf**: è il file che indica su quali porte il server è in ascolto per le richieste in arrivo. Al momento dell'installazione le porte sono quelle standard: 80 per http; 443 per https. Questo file viene richiamato direttamente all'interno di *apache2.conf*.
- **envvars**: in questo file vengono definite le variabili d'ambiente in uso nei file di configurazione.
- **sites-available**: questa cartella racchiude i file di configurazione di tutti i Virtual Host (VH), ovvero tutti i siti presenti sul server. I VH consentono di avere più siti all'interno dello stesso web server come vedremo più avanti.
- **sites-enabled**: questa cartella contiene collegamenti simbolici a *sites-available* e rappresenta la lista dei siti effettivamente accessibili ai client.
- **mods-available**: questa cartella contiene i file di configurazione per i moduli del web server.
- **mods-enabled**: come avviene per *sites-enabled* anche qui sono contenuti i collegamenti simbolici a *mods-available* dei moduli effettivamente abilitati e avviati all'interno del server Apache.

Direttive principali:

- **ServerRoot**: definisce la directory principale con per le configurazioni di Apache. Il valore di default è */etc/apache2*.
- **DocumentRoot**: indica la directory contenente il sito web richiesto al server. Nella nostra distro Ubuntu il valore predefinito è */var/www*.
- **ServerAdmin**: indica l'indirizzo e-mail dell'amministratore del server. Comparirà nei messaggi di errore che i client ricevono.
- **ServerName**: indica l'hostname al quale il server Apache deve rispondere.
- **ServerAlias**: indica nomi alternativi per raggiungere per un host.
- **Listen**: definisce la porta sulla quale il server è in ascolto.
- **User e Group**: definiscono l'utente e il gruppo proprietari del processo.
- **ErrorLog**: indica il percorso dove registrare i file di log degli errori.
- **CustomLog**: indica il percorso dove registrare il file di log degli accessi e il suo formato.
- **LogLevel**: definisce il livello di dettaglio dei log.
- **Allow from**: indica a quali indirizzi IP o hostname è consentito accedere al sito.
- **Deny from**: indica a quali indirizzi IP o hostname non è consentito accedere al sito.
- **PidFile**: il file che contiene il PID del processo padre Apache2.
- **Timeout**: il tempo di inattività dopo il quale Apache2 chiude una connessione client.
- **KeepAlive**: indica se sono permesse o no (On/Off) connessioni persistenti, ovvero in grado di servire più di una richiesta di uno stesso client.
- **MaxKeepAliveRequests**: il massimo numero di richieste che in una singola connessione un client può fare al server. Impostato a 0 le richieste possono essere illimitate.

- **KeepAliveTimeout:** indica il numero di secondi che un client deve attendere prima di effettuare un'altra richiesta su una stessa connessione.