



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

VIRTUALIZZAZIONE TRAMITE DOCKER PRESSO I LABORATORI NAZIONALI DI LEGNARO

Relatore: Prof. Mauro Migliardi

Laureando: Marco Sadocco

ANNO ACCADEMICO 2021-2022

Data di laurea 21-09-2022

Indice

1	Introduzione	3
1.1	Obiettivi	4
2	Luogo di lavoro	5
2.1	Istituto Nazionale di Fisica Nucleare	5
2.2	Laboratori Nazionali di Legnaro	6
2.3	Divisione Ricerca	7
2.4	Servizio Tecnologie Informatiche	7
3	Ambiente di sviluppo	10
3.1	Windows 11	10
3.2	VirtualBox	11
3.3	Visual Studio Code	12
3.4	CentOS 7	12
3.5	Ubuntu	13
4	Virtualizzazione	14
4.1	Virtualizzazione a livello di sistema operativo	14
4.2	Livello Hardware	15
4.2.1	Bare-metal e Hosted	15
4.2.2	Para-virtualization e full-virtualization	16
4.3	Soluzione dei Laboratori Nazionali di Legnaro	16
4.4	Soluzione proposta	16
5	Docker	17
5.1	Vantaggi	17
5.2	Svantaggi	18
5.3	Tecnologie utilizzate ad esso ausiliarie	18
5.3.1	Docker compose	18
5.3.2	I volumi e i bind mounts	19

5.3.3	Traefik	19
6	Implementazione	20
6.1	Infrastruttura	20
6.2	Ubuntu	20
6.3	Proxmox	20
6.4	Applicazioni containerizzate	21
6.4.1	Wordpress	21
6.4.2	Piwigo	21
6.5	Procedimento	21
7	Confronto Prestazioni	23
8	Conclusioni	24
9	Bibliografia	25

1 Introduzione

Questa tesi si basa sul tirocinio da me svolto presso il Servizio Tecnologie Informatiche dei Laboratori Nazionali di Legnaro.

Ad oggi i Laboratori Nazionali di Legnaro sfruttano la full-virtualization sia per applicazioni di carattere scientifico, sia per erogare servizi informatici, ciò comporta la creazione di nuove macchine virtuali per ogni nuovo servizio informatico mediante il server software open source Proxmox¹. Nonostante la facilità di utilizzo e la scalabilità, esso presenta i seguenti punti deboli:

- i servizi in produzione sulla stessa macchina potrebbero necessitare di versioni diverse della stessa libreria;
- l'allocazione delle risorse non è sempre ottimale;
- la configurazione e la manutenzione della macchina virtuale va ripetuta per ogni singolo servizio.

Negli ultimi anni si è vista la diffusione dei container, che migliorano l'accessibilità e facilitano l'utilizzo della virtualizzazione a livello di sistema operativo. I container sono un'evoluzione della para-virtualization, essi hanno varie implementazioni tra cui Docker che, con l'ecosistema relativo, è uno dei software più popolari e promettenti. I potenziali vantaggi dei container sono:

- isolamento tra le applicazioni: ogni applicazione ha le proprie specifiche dipendenze e risorse e non entra in conflitto con le altre, è quindi possibile creare più istanze della stessa applicazione (anche di versioni diverse) senza che esse entrino in conflitto;
- scalabilità: le risorse, come ad esempio cores della CPU e Random Access Memory (RAM), vengono utilizzate solo quando necessario, evitando l'allocazione statica;
- portabilità: per trasferire un servizio da un sistema all'altro è sufficiente copiare un file di testo e non un intero file-system.

¹<https://www.proxmox.com/en/>

La tesi tratterà:

- una descrizione sempre più granulare dell'Istituto Nazionale di Fisica Nucleare, fino a raggiungere il Servizio Tecnologie Informatiche di cui faccio parte (capitolo 3);
- gli strumenti utilizzati per imparare ad utilizzare Docker (capitolo 4);
- i vari tipi di virtualizzazione (capitolo 5);
- il software Docker, le tecnologie ad esso ausiliarie e la comparazione della virtualizzazione a livello di sistema operativo con la full-virtualization (capitolo 6);
- l'implementazione dei servizi mediante Docker (capitolo 7);
- la valutazione dei vantaggi e degli svantaggi che comporta il passaggio dalla full-virtualization alla gestione di alcuni servizi mediante i container (capitolo 8).

1.1 Obiettivi

Il proposito di introdurre nel Servizio Tecnologie Informatiche l'utilizzo dei container è motivato dai grandi vantaggi che essi offrono, si tratta infatti di una tecnologia molto promettente che sta vedendo una sempre maggiore diffusione. Altra motivazione è il già presente utilizzo dei container in altri servizi dei Laboratori, con i quali il Servizio Tecnologie Informatiche collabora.

Per quanto ovvio, valutare la convenienza del passaggio dei servizi informatici da una virtualizzazione di tipo full-virtualization ad una virtualizzazione a livello di sistema operativo è stata la prima attività del tirocinio. Appurata la convenienza di detto passaggio, l'implementazione del nuovo sistema di servizi è stato l'obiettivo primario del mio lavoro, volto soprattutto a rendere più mantenibile, efficace, efficiente e semplice l'infrastruttura dei servizi.

Per raggiungere tali obiettivi, è stato necessario in via preliminare imparare ad utilizzare Docker, uno dei più diffusi gestori di container. Grazie all'utilizzo di Docker sono stati quindi virtualizzati, a livello di sistema operativo, il sito web e il gestore di materiale audio-visivo, che utilizzano come Content Management System (CMS) rispettivamente Wordpress e Piwigo. L'implementazione ha quindi riguardato servizi che erano già in produzione, fruibili sia da utenti esterni che interni ai Laboratori.

2 Luogo di lavoro

2.1 Istituto Nazionale di Fisica Nucleare

L'Istituto Nazionale di Fisica Nucleare (INFN) è un ente pubblico nazionale di ricerca dedicato allo studio dei costituenti fondamentali della materia e delle leggi che li governano. Le sue attività si svolgono negli ambiti teorico e sperimentale della fisica subnucleare, nucleare e astro-particellare.

Le attività di ricerca dell'INFN si svolgono tutte in un ambito di cooperazione internazionale e in stretta collaborazione con il mondo universitario italiano, sulla base di consolidati e pluridecennali rapporti. La ricerca fondamentale in questi settori richiede l'uso di tecnologie e strumenti di ricerca d'avanguardia che l'INFN sviluppa sia nei propri laboratori sia in collaborazione con il mondo dell'industria. L'INFN partecipa alle attività di ricerca del CERN¹, il più grande laboratorio al mondo di fisica delle particelle, che si trova al confine tra Francia e Svizzera. Questa collaborazione nacque negli anni cinquanta del secolo scorso con lo scopo di costruire ed utilizzare macchine acceleratrici sempre più potenti.[1]

L'Istituto è organizzato nelle seguenti strutture: Sezioni, Laboratori Nazionali, Centri Nazionali e Amministrazione Centrale.

- Le Sezioni sono strutture scientifiche con lo scopo di svolgere attività di ricerca e alta formazione, esse hanno spesso sede presso i dipartimenti di fisica delle Università.
- I Laboratori Nazionali sono strutture scientifiche aventi il fine di sviluppare e gestire i complessi strumentali per le attività di ricerca dell'ente, della comunità nazionale e di quella internazionale.
- I Centri Nazionali sono:

¹in francese Conseil Européen pour la Recherche Nucléaire

- strutture scientifico-tecnologiche con il fine di sviluppare e gestire apparecchiature strumentali e di svolgere attività di ricerca e sviluppo tecnologico;
 - strutture scientifico-formative con il fine di promuovere il progresso delle conoscenze scientifiche e di predisporre i giovani alla ricerca scientifica.
- L'Amministrazione Centrale ha i compiti di:
 - gestire le funzioni amministrative centralizzate;
 - svolgere funzioni di indirizzo, coordinamento e verifica dell'attività amministrativa decentrata;
 - assicurare i servizi tecnici, professionali e di monitoraggio centrali;
 - predisporre ed eseguire gli atti deliberativi di competenza.

2.2 Laboratori Nazionali di Legnaro

I Laboratori Nazionali di Legnaro (LNL) sono una delle quattro strutture di supporto alla ricerca dell'INFN. I sei acceleratori presenti permettono studi di fisica e astrofisica nucleare e ricerche interdisciplinari. I Laboratori sono principalmente orientati allo sviluppo e alla innovazione degli acceleratori di particelle, dei rivelatori di radiazioni e delle tecnologie associate. Altra attività di rilievo dei Laboratori consiste nel trasferimento tecnologico che sfrutta le competenze dei Laboratori quali le tecnologie di trattamento delle superfici e la deposizione di film sottili. In futuro è prevista anche la produzione di radioisotopi di interesse medico.

La struttura dei Laboratori Nazionali di Legnaro è composta da servizi direttamente afferenti al Direttore: Amministrazione, Direzione, Fondi Esteri, Radioprotezione, Prevenzione e protezione, Sicurezza Non Convenzionale. Ai servizi si aggiungono la divisione Ricerca, la divisione Acceleratori e la divisione Tecnica e dei Servizi Generali. Ogni divisione è strutturata in servizi.



Figura 2.1: logo dei Laboratori Nazionali di Legnaro[3]

2.3 Divisione Ricerca

La divisione ricerca ha i seguenti compiti:

- gestire e sviluppare i supporti tecnico-scientifici necessari alla ricerca e alla sperimentazione;
- collaborare allo sviluppo, alla gestione e alla realizzazione dei complessi strumentali dei Laboratori Nazionali di Legnaro;
- promuovere gli esperimenti dei Laboratori Nazionali di Legnaro, i progetti internazionali e la diffusione nel mondo produttivo delle conoscenze e delle tecnologie acquisite;
- gestire la comunicazione scientifica e promuovere l'immagine dei laboratori nel territorio.

I seguenti servizi compongono la divisione ricerca:

- Servizio Comunicazione Scientifica e Documentazione;
- Servizio Radioisotopi per la Fisica Nucleare;
- Servizio Bersagli;
- Servizio Utenti;
- Servizio Tecnologie Informatiche;
- Servizio Tecnologia delle Superfici e Superconduttività;
- Servizio Radioisotopi per la Medicina e la Fisica Applicata.[2]

2.4 Servizio Tecnologie Informatiche

Il Servizio Tecnologie Informatiche a cui afferisco e che fa parte della Divisione Ricerca detta Tier 2, supporta utenti interni ed esterni, gestisce l'infrastruttura del calcolo scientifico per gli esperimenti locali e per le collaborazioni CMS², ALICE³ al LHC⁴ del CERN. Cura inoltre l'organizzazione della rete dei Laboratori e il mantenimento in esercizio dei sistemi di acquisizione dati

²Compact Muon Solenoid

³A LHC Ion Collider Experiment

⁴Large Hadron Collider

per gli esperimenti di fisica nucleare. Il Servizio inoltre è in stretta collaborazione con l'INFN CNAF⁵ e contribuisce alle infrastrutture INFN comuni di supporto per le attività come business continuity e disaster recovery.[4]

⁵Centro Nazionale Analisi Fotogrammi, il centro di calcolo principale dell'INFN

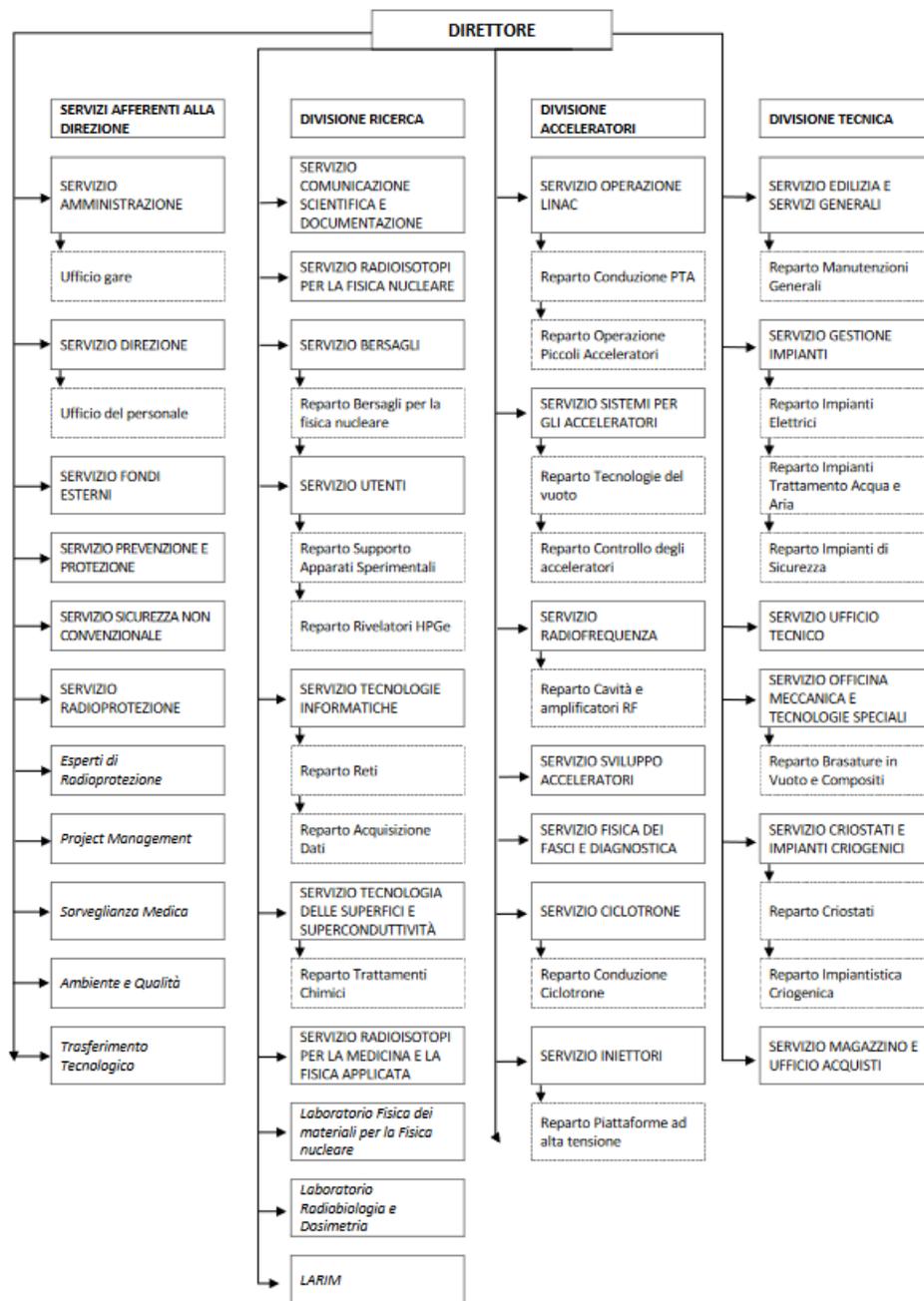


Figura 2.2: Organigramma dei Laboratori Nazionali di Legnaro[5]

3 Ambiente di sviluppo

In questo capitolo vengono illustrati i software adoperati per imparare ad utilizzare Docker: il sistema operativo host, l'editor di testo, il software con cui ho creato e gestito le macchine virtuali di prova ed i sistemi operativi di dette macchine.

3.1 Windows 11

Il computer utilizzato per sperimentare e lavorare con Docker impiegava Windows 11 come sistema operativo. Windows 11 è progettato per personal computer, è un prodotto dalla Microsoft Corporation e fa parte della famiglia di sistemi operativi Windows NT¹ che è la più diffusa nel mondo dei personal computer, come illustrato nel grafico della figura 3.1.



Figura 3.1: logo Windows 11[10]

Windows 11 è un sistema operativo multitasking, permette quindi di eseguire più programmi contemporaneamente; è inoltre multiutente e multiprocessore.[6][7][8]

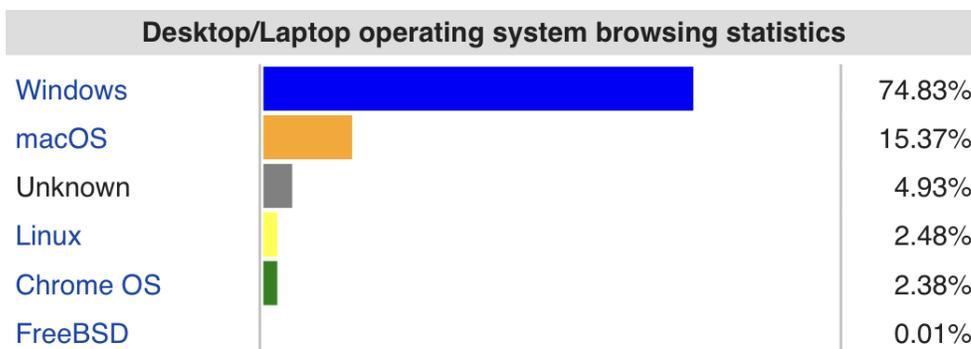


Figura 3.2: sistemi operativi desktop nell'aprile 2022 secondo StatCouner[9]

¹Windows New Technology, la famiglia di sistemi operativi prodotti da Microsoft, unica linea di sistemi operativi per personal computer

3.2 VirtualBox

Oracle VM VirtualBox è un software gratuito e open source che permette l'esecuzione di macchine virtuali per architetture x86 e 64bit. Con VirtualBox è possibile configurare, creare e usare più sistemi operativi (guest) nel proprio PC usando un solo sistema operativo (host). Ogni sistema operativo risiede in una macchina virtuale. I sistemi operativi che VirtualBox supporta come host sono Windows, GNU/Linux e macOS, mentre può eseguire Windows, GNU/Linux, OS/2 Warp, BSD, FreeBSD e infine Solaris e OpenSolaris come sistemi operativi guest.

Si possono configurare per ogni macchina virtuale hardware e periferiche con ogni configurazione indipendente dalle altre.

Ogni macchina guest può essere avviata, fermata o spenta indipendentemente dalle altre. Inoltre ogni stato delle macchine guest può essere salvato come file. Tali files possono poi essere caricati ed eseguiti in altri host indipendentemente dal sistema operativo installato, l'unica condizione necessaria è la presenza di VirtualBox nella macchina host.[12]

Questo software è stato utilizzato inizialmente per creare una macchina virtuale con lo scopo puramente didattico di imparare ad utilizzare Docker. È stata successivamente creata una seconda macchina virtuale che è stata utilizzata per verificare il corretto funzionamento della nuova implementazione del sito Wordpress e di Piwigo. L'unica modifica apportata alla configurazione standard della seconda macchina è stata l'aggiunta di tre schede di rete, rispetto alla assegnazione di default di una singola. Le funzioni delle quattro schede di rete sono state:

- scheda 1 (quella già presente): fornire una connessione Internet alla macchina virtuale,
- scheda 2: utilizzare l'IP assegnato per poter vedere l'interfaccia fornita dal reverse-proxy utilizzato,
- scheda 3: utilizzare l'IP assegnato per poter vedere il sito web dalla macchina host,
- scheda 4: utilizzare l'IP assegnato per poter vedere il gestore di contenuti multimediali dalla macchina host.



Figura 3.3: logo VirtualBox [11]

Per la prima scheda la configurazione utilizzata era quella di default: una connessione NAT che permette alla macchina guest di collegarsi mediante la macchina host alla rete. Per le altre tre schede è stata utilizzata la configurazione host-only, grazie alla quale la macchina guest è connessa solamente alla macchina host ed alle altre eventuali macchine virtuali.

3.3 Visual Studio Code

Per utilizzare la linea di comando, gestire e modificare i files di configurazione, il tutto dalla macchina host, è stato utilizzato il software open source Visual Studio Code, sviluppato da Microsoft e disponibile per Windows, Linux e macOS.

Esso permette l'installazione di molteplici estensioni che ampliano le funzionalità dell'editor[13], ad esempio aggiungendo l'auto completamento per i linguaggi di programmazione utilizzati, aggiungendo strumenti per eseguire test dei programmi o strumenti per compilare ed eseguire i files sorgente in maniera automatica. Nello specifico io ho fatto uso della estensione Remote Development, la quale permette di aprire qualsiasi cartella in una macchina remota, permettendo di utilizzare il terminale del computer a cui si è connessi e di modificare i files di testo presenti al suo interno facendo uso di tutte le funzionalità disponibili in Visual Studio Code.

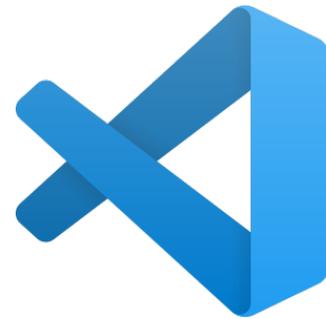


Figura 3.4: logo Visual Studio Code[14]

3.4 CentOS 7

CentOS² è una distribuzione Linux concepita, sviluppata e utilizzata per gli ambienti server.

Deriva dal codice sorgente di Red Hat Enterprise (software open source ma a pagamento), dal quale gli sviluppatori prendono spunto con il fine di rendere CentOS un'alternativa gratuita. La differenza principale tra Red Hat e CentOS è che quest'ultima è priva di assistenza tecnica, motivo per cui Red Hat è a pagamento. Altra differenza risiede nella garanzia, dato che su CentOS le applicazioni finali eseguibili



Figura 3.5: logo CentOS[16]

²Acronimo di Community ENTerprise Operating System

non vengono certificate, a differenza di quanto accade con Red Hat.[15]

Nella mia sperimentazione ho inizialmente utilizzato CentOS 7, versione che è stata distribuita il 7 luglio 2014. Questa scelta è dovuta al fatto che al momento CentOS 7 è il sistema operativo utilizzato nella maggior parte dei server dei Laboratori Nazionali di Legnaro.

Successivamente ho abbandonato l'utilizzo di questo sistema operativo dato che la fine del supporto per gli aggiornamenti relativi alla sicurezza di CentOS 7 è stata stabilita per il 30 giugno 2024. In tale previsione i Laboratori Nazionali di Legnaro hanno in programma la scelta di un nuovo sistema operativo da utilizzare per i server ed attualmente il principale candidato è Ubuntu Server 20.04.

3.5 Ubuntu

Ubuntu è un sistema operativo nato nel 2004 e basato su Linux, più precisamente sul ramo instabile di Debian. È prevalentemente composto da software open source con licenza GNU GPL, ma supporta anche software proprietario.[18]

Ubuntu prevede un nuovo rilascio con supporto a lungo termine (LTS³) ogni due anni e rilasci a breve supporto ogni sei mesi.

È una delle distribuzioni Linux più facili da usare e maggiormente simile all'ambiente di lavoro degli sviluppatori. Inoltre Ubuntu 20.04 LTS Server presenta un termine del supporto di sicurezza fissato per aprile 2030. Queste tre caratteristiche hanno reso Ubuntu la distribuzione scelta dai Laboratori Nazionali di Legnaro per mettere in produzione il sito web e il gestore di contenuti multimediali e conseguentemente è il sistema operativo che ho adoperato per realizzare la virtualizzazione di questi due servizi.



Figura 3.6: logo Ubuntu[17]

³Long Term Service

4 Virtualizzazione

La virtualizzazione è la possibilità di astrarre le componenti hardware degli elaboratori al fine di renderle disponibili al software in forma di risorsa virtuale. Questo insieme di risorse virtuali forma le macchine virtuali, sulle quali possono essere installati sistemi operativi e relative applicazioni.

Le risorse virtuali possono essere mappate ad un insieme di risorse che possono essere fisiche o virtuali nel caso di più livelli di virtualizzazione.

Le principali tecniche di virtualizzazione sono:

- isolamento: le risorse virtuali sono presentate senza dipendenze dal sottosistema fisico, permettendo di sostituire l'hardware senza modifica delle risorse virtuali;
- emulazione: le risorse virtuali hanno una diversa interfaccia da quelle fisiche;
- condivisione: più risorse virtuali vengono associate alla stessa risorsa fisica;
- aggregazione: più risorse fisiche vengono mappate alla stessa risorsa virtuale, in modo tale da unirne le caratteristiche.[19]

Esistono diverse tipologie differenti di virtualizzazione che possono essere divise come nello schema in figura 4.1.

4.1 Virtualizzazione a livello di sistema operativo

La virtualizzazione a livello di sistema operativo è un metodo di esecuzione delle applicazioni che isola le risorse hardware e software utilizzate da esse rimanendo sempre all'interno di un solo e unico sistema operativo, in questo modo l'applicazione può vedere solo le risorse che le sono state assegnate.

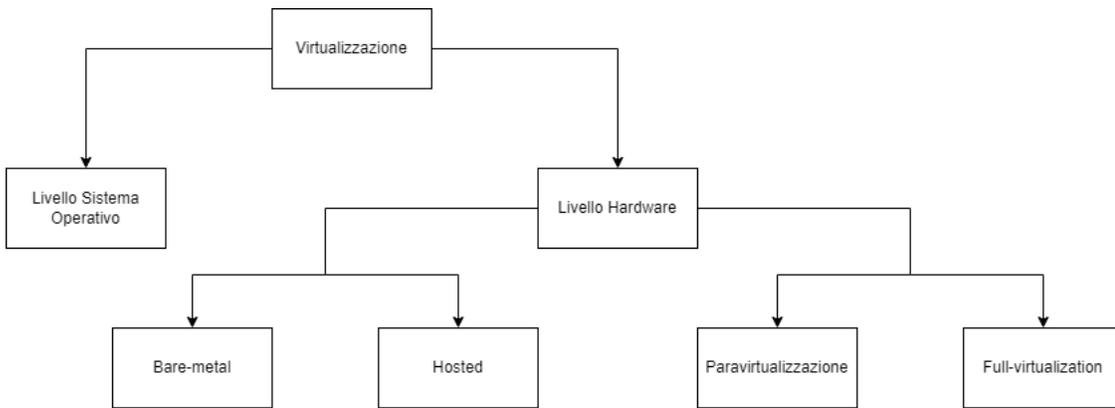


Figura 4.1: Tipi di virtualizzazione

Tutto ciò viene realizzato mediante la creazione di container: istanze user-space isolate tra di loro. In questo tipo di virtualizzazione sono assenti le macchine virtuali ed i rispettivi sistemi operativi guest.[21]

4.2 Livello Hardware

Nella virtualizzazione hardware i sistemi operativi vengono eseguiti in modo concorrente sullo stesso hardware pur essendo eterogenei. In questo tipo di virtualizzazione è presente l'hypervisor che ha il compito di presentare all'utente i sistemi operativi delle macchine guest e di gestire la loro esecuzione. Un altro suo compito è quello di gestire l'accesso alle risorse hardware e garantire la protezione e l'isolamento tra le macchine.

4.2.1 Bare-metal e Hosted

La virtualizzazione a livello hardware può essere suddivisa in due categorie in base alla presenza o meno del sistema operativo:

- bare-metal: il sistema operativo host è assente e le sue funzioni vengono sostituite dall'hypervisor;
- hosted: l'hypervisor è un normale processo utente sul sistema operativo host.

4.2.2 Para-virtualization e full-virtualization

Le macchine virtuali possono essere suddivise in base a come esse si interfacciano all'hardware, possono quindi utilizzare la para-virtualization oppure la full-virtualization.

- Para-virtualization: il sistema di virtualizzazione espone ad ogni macchina virtuale interfacce hardware simulate simili alle corrispondenti interfacce fisiche. Queste interfacce sono API che implementano una astrazione delle periferiche. I sistemi operativi guest vanno quindi modificati per renderli compatibili con queste periferiche.[22]
- Full-virtualization: le macchine virtuali hanno la stessa interfaccia di una macchina fisica. In questo modo il sistema operativo guest non deve essere modificato per poter comunicare con l'hardware.[20]

4.3 Soluzione dei Laboratori Nazionali di Legnaro

Attualmente nei Laboratori Nazionali di Legnaro molti dei servizi informatici sono ospitati in un unico cluster. Essi vengono virtualizzati mediante la tecnica della full-virtualization basata su di un sistema bare-metal che impiega come hypervisor Proxmox il quale implementa il modulo KVM, il piú diffuso nell'ambito della virtualizzazione open source.

I Laboratori hanno scelto questa metodologia per isolare in maniera piú efficiente ogni servizio, sfruttando al meglio l'hardware a disposizione dei Laboratori. Inoltre avendo un unico cluster ne risulta una manutenzione piú semplice e duttile. Questa implementazione infine permette anche una maggiore availability.

4.4 Soluzione proposta

La soluzione proposta è quella di raggruppare alcuni servizi omogenei tra di loro in specifiche macchine virtuali. Per fare ciò si sfrutterà la infrastruttura di cui sopra utilizzando Docker come gestore dei container. Inoltre con questa nuova implementazione si ridurranno sia lo spreco di risorse dovuto ai multipli sistemi operativi guest sia il numero di macchine virtuali da mantenere, tenendo comunque le macchine virtuali isolate. Il tutto risulterà in un aumento della efficienza di gestione.

5 Docker

Docker è un software open source progettato per eseguire e gestire i container, con lo scopo di facilitare i processi di sviluppo delle applicazioni. Si basa sulla virtualizzazione a livello di sistema operativo, nella quale un singolo sistema operativo si occupa di orchestrare l'isolamento e le limitazioni delle risorse.[23]

5.1 Vantaggi

- **Isolamento:** memoria, processi, dispositivi e rete sono virtualizzati a livello di sistema operativo e le applicazioni sono isolate l'una dall'altra. Non ci sono quindi conflitti di dipendenze o risorse contese, dato che ogni applicazione ha dei limiti definiti di utilizzo delle risorse. Tutto ciò risulta anche in una protezione maggiore.
- **Sviluppo semplificato:** ogni container spesso include non solo l'applicazione/servizio ma anche tutto il pacchetto utile per eseguirlo, semplificando ogni operazione di deployment e facilitando la distribuzione su differenti sistemi operativi senza ulteriori configurazioni.
- **Tempi di avvio più rapidi:** essendo assente un sistema operativo guest i tempi di avvio dei servizi si accorciano notevolmente.
- **Ambiente uniforme:** i container possono essere distribuiti su qualunque risorsa di calcolo indipendentemente da configurazioni, sistema operativo o hardware, a patto che nel nuovo host sia presente lo stesso gestore di container.
- **Efficienza operativa:** i container possono eseguire più applicazioni sulla medesima istanza specificando le risorse che devono essere usate, in questo modo si ottimizza l'utilizzo delle stesse. Inoltre utilizzano una quantità minore di risorse rispetto alle macchine virtuali vista la presen-

za di un unico sistema operativo per tutti i servizi e non uno per ogni applicazione.

- Migliore manutenibilità: avendo una sola macchina virtuale da gestire, con tutti i container al suo interno, la manutenzione risulta semplificata dato che aggiornamenti di sistema e modifiche alla configurazione vanno eseguite una singola volta.[24][25]

5.2 Svantaggi

- Prestazioni: i container consumano meno risorse delle macchine virtuali ma allo stesso tempo sono soggetti al dispendio di risorse dovuto alla rete accessoria che viene fornita ai container per comunicare tra di loro e con la macchina host.
- Isolamento peggiore rispetto alla virtualizzazione a livello hardware: dato che i container non contengono un proprio sistema operativo i processi in esecuzione non sono isolati in maniera perfetta.[26]

5.3 Tecnologie utilizzate ad esso ausiliarie

In questa sezione sono descritte le tecnologie utilizzate assieme a Docker: docker-compose, i sistemi di gestione dati di Docker e il reverse proxy Traefik.

5.3.1 Docker compose

Docker compose è uno strumento che permette di definire e avviare applicazioni Docker multi container mediante un file YAML che contiene tutte le indicazioni necessarie. Con un singolo comando è quindi possibile avviare contemporaneamente tutti i servizi che sono stati configurati nello stesso file. Con la linea di comando invece è possibile far funzionare un solo container alla volta, motivo per cui Docker Compose risulta più efficiente per la creazione di applicazioni multi-container. Altro vantaggio nel suo utilizzo è il mantenere segnate tutte le configurazioni in un file, potendo quindi tracciare facilmente ciò che è stato creato, avendo quindi una sorta di documentazione dei propri container. Infine dato che il file contenente tutta la configurazione rimane persistente, è possibile effettuare piccole modifiche allo stesso per testare diverse configurazioni dei container.

5.3.2 I volumi e i bind mounts

I volumi ed i bind mounts sono delle tecnologie che permettono di creare dati persistenti generati ed usati dai container. I volumi sono creati e gestiti completamente da Docker. I bind mounts invece sono delle mappature tra le cartelle delle macchine host con il container, nel caso la cartella non esista nel sistema host essa viene creata al momento. I bind mounts sono molto performanti e dipendono dalla struttura del file system della macchina.

5.3.3 Traefik

Traefik è un reverse proxy open source, si integra perfettamente con Docker dato che esso stesso è un container, ciò gli permette di auto-configurarsi il routing. La sua configurazione si basa su un file che contiene la configurazione statica e uno contenente la configurazione dinamica. Affinché avvengano le modifiche apportate alla prima è necessario riavviare il container del reverse proxy, mentre nel caso si modifichi la seconda i cambiamenti prenderanno subito piede. Traefik gestisce i container mediante delle label che oltre a fornire istruzioni per il routing permettono di aggiungere middlewares.



Figura 5.1: logo Traefik[27]

6 Implementazione

6.1 Infrastruttura

La macchina virtuale è stata creata nel cluster contenente i servizi informatici dei laboratori, ha quattro CPU, quattro gibi-byte di RAM ed un disco di sistema di cinquanta gibi-byte. Ad essa sarà aggiunto un disco dati che avrà lo scopo di contenere i file multimediali dei due servizi.

6.2 Ubuntu

Come sistema operativo della macchina virtuale è stato utilizzato Ubuntu Server LTS 20.04. Su di esso è poi stato eseguito uno script realizzato nei Laboratori Nazionali di Legnaro, il quale integra nella macchina virtuale determinati servizi, che la rendono più sicura e più facile da utilizzare per gli utenti dei Laboratori. Degli esempi sono l'installazione automatica di Docker e Docker Compose e l'autenticazione mediante le credenziali INFN LNL.

6.3 Proxmox

Proxmox Virtual Environment (Proxmox VE o PVE) è un software open source per la gestione della virtualizzazione. È una distribuzione Linux basata su Debian con un kernel Ubuntu LTS, modificato in modo tale da essere il più leggero possibile. Consente l'implementazione e la gestione di macchine virtuali. Include una console web per gestirle ed altri strumenti utilizzabili mediante la riga di comando.

Proxmox può essere raggruppato su più nodi server. In questo modo se un nodo Proxmox diventa non disponibile o non funziona, i server virtuali possono essere spostati automaticamente su un al-



Figura 6.1: logo Proxmox[28]

tro nodo e riavviati. È possibile inoltre spostare le macchine virtuali live da un host fisico ad un altro senza tempi di inattività. Infine include il software di backup, v2dumpp, che consente la compressione dati e il funzionamento in modalità in linea o snapshot. Esiste anche Proxmox Backup Server, un software ad architettura client-server dedicato ai backup che offre compressione, cifratura autenticata e backup incrementali.[29]

6.4 Applicazioni containerizzate

6.4.1 Wordpress

Wordpress è un content management system (CMS) multiautore open source che permette di gestire e creare un sito Internet formato da contenuti testuali o multimediali gestibili in maniera dinamica. È sviluppato in PHP e si basa sul gestore di database MariaDB. Permette la creazione di pagine statiche o di articoli che possono essere raggruppati in categorie. Le sue funzionalità possono essere estese tramite plug-in. Dispone di temi per personalizzare il proprio sito. Utilizza degli URL permanenti per aiutare l'ottimizzazione nei motori di ricerca.[31]



Figura 6.2: logo Wordpress[30]

6.4.2 Piwigo

Piwigo, il cui precedente nome era PhpWebGallery, è un'applicazione web che permette di creare e condividere una galleria fotografica online. È un software open source scritto in PHP e necessita la presenza di un database (MySQL, PostgreSQL o SQLite). L'immagine docker utilizzata è stata sviluppata da Linuxserver, una community volta allo sviluppo di immagini docker open source.[33]



Figura 6.3: logo Piwigo[32]

6.5 Procedimento

Inizialmente è stata creata una macchina virtuale con Proxmox. Su di essa è stato installato Ubuntu Server LTS, modificato poi con il template realizzato dai Laboratori Nazionali di Legnaro, con

il quale sono stati installati Docker e Docker Compose. Sono stati poi creati i seguenti container: Traefik, Wordpress, Piwigo e due istanze di MariaDB, una per il sito web e una per la galleria online. Nell'installazione dei servizi si è anche fatto use dei bind mounts per rendere i dati dei database, delle configurazioni e dei contenuti multimediali permanenti. Al DNS dei Laboratori sono stati aggiunti gli indirizzi di Wordpress e del gestore di contenuti multimediali. A questi due container sono state aggiunte le label necessarie per gestirli mediante Traefik. Per quanto riguarda quest'ultimo è stato scelto di non esporre la sua interfaccia grafica di default, qualora si dovesse accedere è necessario modificare la configurazione statica di Traefik e riavviare il suo container.

7 Confronto Prestazioni

Nella tabella 7.1 sono raffigurate le risorse assegnate (RAM e Cores della CPU) alle macchine contenenti Wordpress, Piwigo ed i servizi containerizzati. L'ultima riga infine contiene la somma delle risorse delle prime due macchine.

Macchina	Core assegnati	RAM assegnata (GiB)
Wordpress	4	2
Piwigo	4	4
Container	4	4
Wordpress + Piwigo	8	6

Tabella 7.1: risorse assegnate

Si può notare come, grazie all'utilizzo dei container, la riduzione del numero di sistemi operativi guest permetta un risparmio di risorse mantenendo comunque i servizi isolati. Nello specifico il si è dimezzato il numero di core assegnati e si è ridotto di un terzo la RAM utilizzata.

8 Conclusioni

La soluzione proposta, nonostante i servizi containerizzati siano soltanto due, necessita già di una minore quantità di risorse rispetto alla implementazione precedente. A questo elemento va aggiunta la forza lavoro che verrà risparmiata dato che sarà sufficiente mantenere una singola macchina virtuale. Con il passaggio graduale di altri servizi dalla virtualizzazione a livello hardware a quella a livello di sistema operativo questi due vantaggi continueranno a valere sempre di più, confermando la bontà di questa tipologia di virtualizzazione. Altro beneficio sarà la semplicità di sviluppo di nuovi servizi visto che le immagini dei container contengono la maggior parte delle dipendenze di cui i servizi necessitano e quindi, nella maggior parte dei casi, la loro messa in produzione viene eseguita mediante un singolo comando dal terminale. La containerizzazione renderà inoltre più semplice una eventuale migrazione dei container, visto che questi ultimi possono essere creati facilmente su più piattaforme hardware e sistemi operativi diversi. Infine adottare questa nuova tecnologia porterà i Laboratori ad avere tempi di testing e produzione minori, grazie alla facilità con cui si possono gestire i container.

9 Bibliografia

- [1] Descrizione dell'Istituto Nazionale di Fisica Nucleare, visitato il 6/7/2022:
<https://www.lnl.infn.it/>
- [2] Statuto dell'Istituto Nazionale di Fisica Nucleare, visitato il 6/7/2022:
<https://www.ac.infn.it/normativa/REGOLAMENTI%20NUOVI%20DA%20PUBBLICARE/REGOLAMENTI/Regolamento%20di%20Organizzazione%20e%20Funzionamento%20dell'INFN.pdf>
- [3] Logo Laboratori Nazionali di Legnaro, visitato il 18/7/2022:
<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRNwbw7eV4Os6oGNdAEtfHoDGuWYko8skXKGQ&usqp=CAU>
- [4] Pagina Servizio Tecnologie Informatiche LNL, visitato il 6/7/2022:
<https://www.lnl.infn.it/servizio-tecnologie-informatiche>
- [5] Statuto dei Laboratori Nazionali di Legnaro, visitato il 6/7/2022:
<https://www.lnl.infn.it/wp-content/uploads/Disciplinare.pdf>
- [6] Windows 11, visitato il 15/7/2022:
https://it.wikipedia.org/wiki/Windows_11
- [7] Microsoft Windows, visitato il 15/7/2022:
https://it.wikipedia.org/wiki/Microsoft_Windows
- [8] Windows NT, visitato il 18/7/2022:
https://it.wikipedia.org/wiki/Windows_NT
- [9] Usage share of operating system, visitato il 15/7/2022:
https://en.wikipedia.org/wiki/Usage_share_of_operating_systems
- [10] Windows 11 logo, visitato il 18/7/2022:
<https://logodownload.org/windows-11-logo/>
- [11] VirtualBox logo, visitato il 18/7/2022:
https://cdn.windowsreport.com/wp-content/uploads/2019/05/Virtualbox_logo.png
- [12] VirtualBox, visitato il 18/7/2022:
<https://it.wikipedia.org/wiki/VirtualBox>
- [13] Visual Studio Code, visitato il 18/7/2022:
https://it.wikipedia.org/wiki/Visual_Studio_Code

[14]VSCode logo, visitato il 18/7/2022:
<https://user-images.githubusercontent.com/674621/71187801-14e60a80-2280-11ea-94c9-e56576f76baf.png>

[15]CentOS, visitato il 20/07/2022:
<https://it.wikipedia.org/wiki/CentOS>

[16]CentOS logo, visitato il 20/07/2022:
https://static-eu.insales.ru/images/products/1/4730/259076730/cent_7.png

[17]Ubuntu 20.04 logo, visitato il 20/07/2022:
<https://meterpreter.org/ubuntu-20-04-lts-final-beta-releases/>

[18]Ubuntu, visitato il 21/07/2022:
<https://it.wikipedia.org/wiki/Ubuntu>

[19]Virtualizzazione, visitato l'8/8/2022:
<https://it.wikipedia.org/wiki/Virtualizzazione>

[20]Tecnologie di virtualizzazione, visitato il 2/08/2022:
https://it.wikiversity.org/wiki/Tecnologie_di_virtualizzazione

[21]Virtualizzazione a livello di sistema operativo, visitato il 29/07/2022:
https://it.wikipedia.org/wiki/Virtualizzazione_a_livello_di_sistema_operativo

[22]Paravirtualizzazione, visitato il 22/08/2022:
<https://it.wikipedia.org/wiki/Paravirtualizzazione>

[23]Docker, visitato il 2/08/2022:
<https://it.wikipedia.org/wiki/Docker>

[24]I vantaggi dei container, visitato il 2/08/2022:
<https://www.criticalcase.com/it/blog/i-vantaggi-dei-container.html>

[25]Vantaggi dei container, visitato il 24/08/2022:
<https://www.netapp.com/it/devops-solutions/what-are-containers/>

[26]Svantaggi Docker, visitato il 3/08/2022:
<https://www.channelfutures.com/open-source/when-not-to-use-docker-understanding-the-limitations-of-containers>

[27]Traefik logo, visitato il 5/08/2022:
<https://traefik.io/>

[28]Proxmox logo, visitato il 19/08/2022:
<https://www.rackone.it/proxmox/>

[29]Proxmox, visitato il 19/08/2022:
https://it.wikipedia.org/wiki/Proxmox_Virtual_Environment

[30]Wordpress logo, visitato il 19/08/2022:
<https://it.wikipedia.org/wiki/WordPress.com>

[31]Wordpress, visitato il 19/08/2022:
<https://it.wikipedia.org/wiki/WordPress>

[32]Piwigo logo, visitato il 19/08/2022:

<https://github.com/Piwigo>

Piwigo, visitato il 20/08/2022:

<https://it.wikipedia.org/wiki/Piwigo>

[33]Piwigo, visitato il 20/08/2022:

<https://it.wikipedia.org/wiki/Piwigo>