



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Dipartimento di Informatica
Tecnologie per la Produzione del Software

**Analisi della tecnologia dei container in favore
del deploy delle applicazioni**

Relatore:

Prof. MICHELE SCALERA

Laureando: Magno Luis Benitez Murillo

Di cosa
parleremo

- VIRTUALIZZAZIONE
- HYPERVISOR
- DOCKER
- DOCKER SWARM
- SPERIMENTAZIONE
- RISULTATI
- SVILUPPI FUTURI

VIRTUALIZZAZIONE

- La virtualizzazione è un meccanismo che permette di eseguire contemporaneamente più sistemi operativi su una sola macchina astruendo le componenti hardware.

• SISTEMA OPERATIVO



• RISORSE



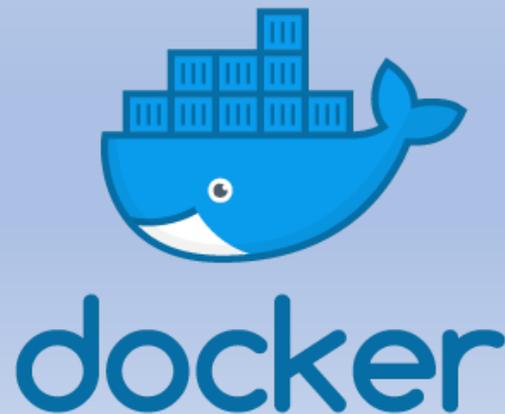
HYPERVISOR



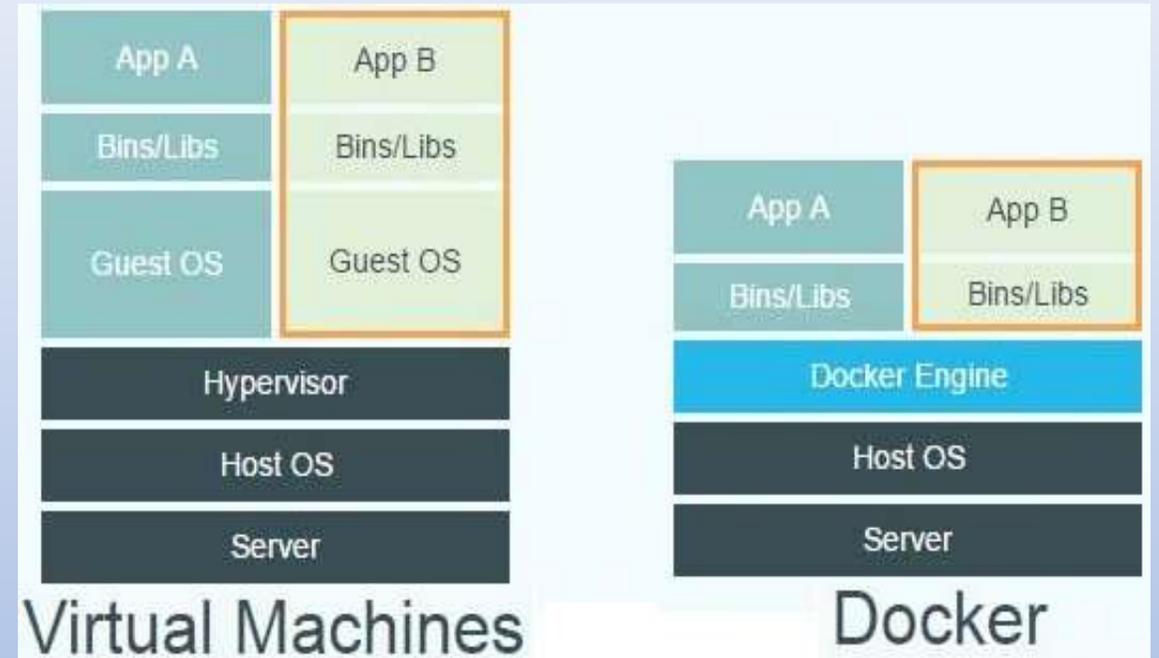
Presenta i sistemi operativi guest con una piattaforma operativa virtuale e gestisce l'esecuzione dei sistemi operativi guest; in questo modo istanze multiple di una varietà di sistemi operativi possono condividere le risorse hardware virtualizzate.

CONTAINERS

- Forma di server virtualizzato a livello del sistema operativo.
- Condividono lo stesso sistema operativo
- Possibilità di pacchettizzare un'applicazione rendendola trasportabile e disponibile verso altri sistemi
- Sicurezza



DOCKER ENGINE



Piattaforma per lo sviluppo che sfrutta i container per il trasporto, e l'esecuzione di applicazioni ed è progettata per fornire applicazioni più velocemente.

COMPONENTI DOCKER

- **IMMAGINI**

Le immagini contengono il filesystem del Sistema Operativo ed una certa applicazione, similmente a una classe Java o un template di una Virtual Machine

- **DOCKER-HUB**

I registri Docker sono la componente di distribuzione e servono per ospitare le immagini:

<https://hub.docker.com>

- **DOCKER FILE**

Costruisce una immagine, definisce l'applicazione e fornisce a Docker le istruzioni su come deve essere definita all'interno dell'immagine.

- **DOCKER-COMPOSE**

Composizione di uno o più containers, l'avvio di questo unico file comporta l'impostazione e l'avvio di tutti i containers presenti al suo interno.

DOCKER SWARM

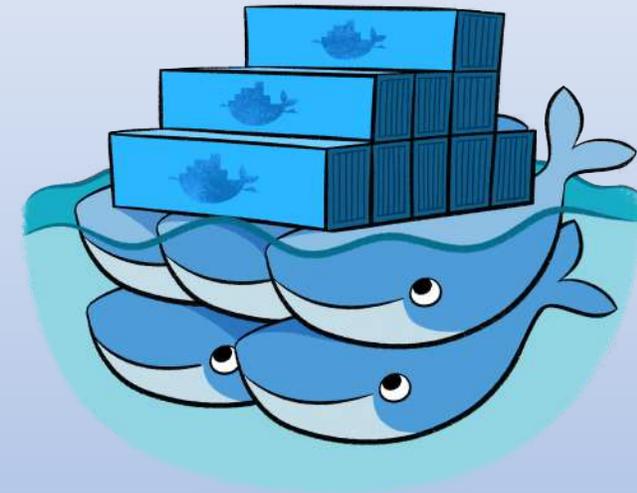
Uno Swarm è un gruppo di macchine che utilizzano Docker e si uniscono in un cluster, Docker Swarm è la piattaforma di orchestrazione di contenitori open source ed il motore cluster nativo di Docker.

- **SWARM MANAGER**

Possono esser più di uno e sono le uniche macchine in uno Swarm che possono eseguire i comandi o autorizzare altri nodi a unirsi allo Swarm come i Worker.

- **WORKER**

I worker vengono usati per fornire capacità e non hanno l'autorità di dire a nessun'altra macchina cosa può o cosa non può fare.



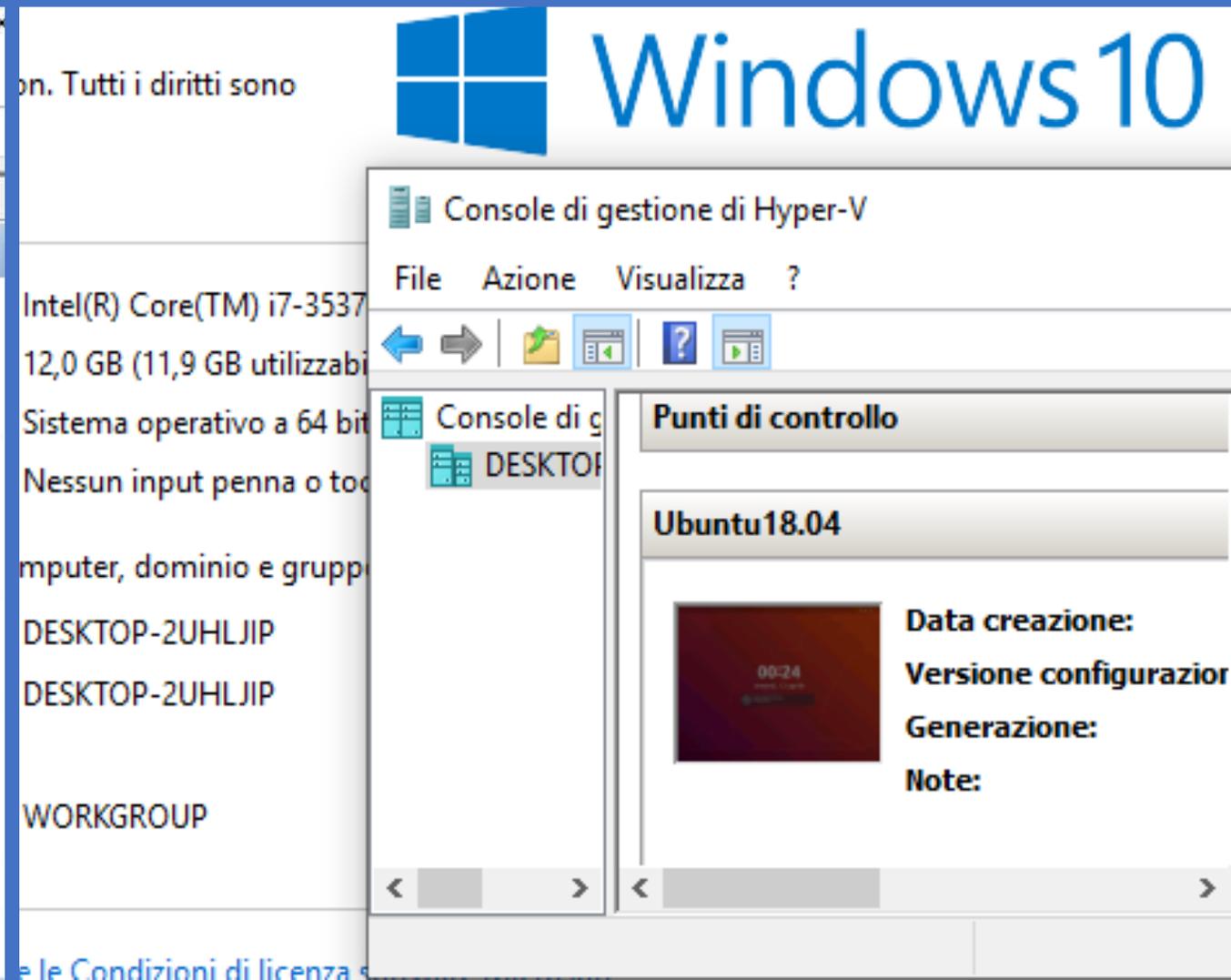
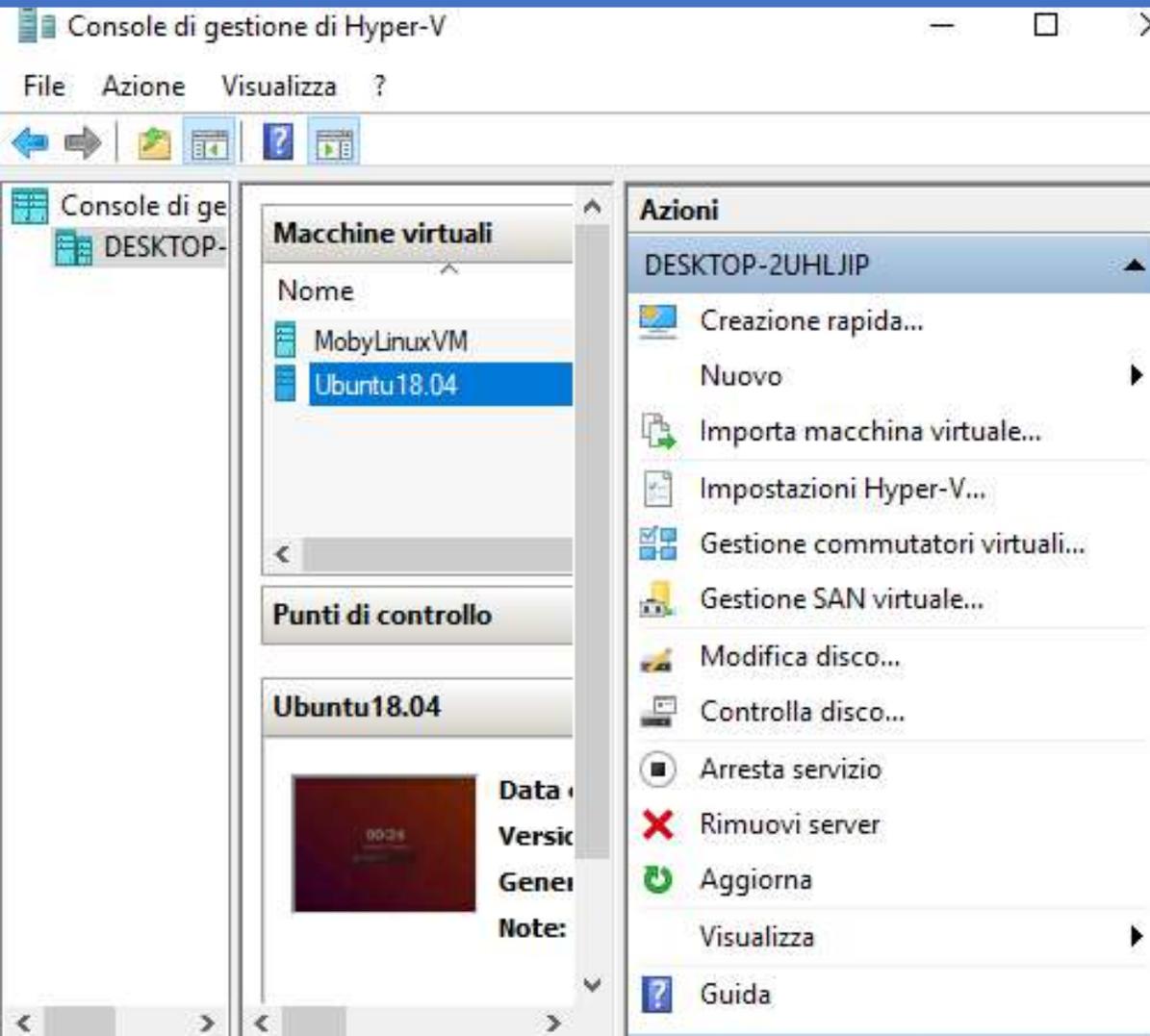
SPERIMENTAZIONE

- NODO MANAGER

Virtual machine Ubuntu 18.04 ls

- NODO WORKER

Macchina fisica Windows 10 pro



SPERIMENTAZIONE

Il comando Docker che permette a una macchina fisica o virtuale di configurare Docker Swarm è: «***docker swarm init***»

Precondizioni:

- Installazione Docker
- Appartenente a una rete di elaboratori

```
root@luis-Virtual-Machine:/home/luis# docker swarm init
Swarm initialized: current node (5w4zpykkjy40imjdaci254uti) is now a
manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-27cl1k5mz5n9qoq6fkzgo2t6gs1qu
hzqj1mdv3vnwgupd9a3me-e1qcbha09wth8ri5vvbvev55z 192.168.229.166:2377

To add a manager to this swarm, run 'docker swarm join-token manager
' and follow the instructions.

root@luis-Virtual-Machine:/home/luis#
```

CONFIGURAZIONE WORKER

```
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> docker swarm join --token SWMTKN-1-27cl1k5mz5n9qoq6fkzgo2t
6gs1quhzqj1mdv3vnwgupd9a3me-e1qcbha09wth8ri5vvbvev55z 192.168.229.166:2377
This node joined a swarm as a worker.
PS C:\WINDOWS\system32>
```

DOCKER-COMPOSE.YAML

- **Services:**

1. Web
2. Db
3. Phpmyadmin

- **Build Dockerfile**

1. Webdevops/php-apache:alpine-php7 -> myimageweb
2. Mysql -> myimagedb
3. Phpmyadmin->myimagephpadmin

- **Port mapping**

1. 4500:80
2. 8088:80

```
version: "3"
```

```
services:
```

```
  web:
```

```
    image: myimageweb
```

```
    build: ./web
```

```
    volumes:
```

```
      - ./web/www:/app
```

```
    ports:
```

```
      - "4500:80"
```

```
  db:
```

```
    image: myimagedb
```

```
    build: ./db
```

```
    command: --default-authentication-plugin=mysql_native_password
```

```
    restart: always
```

```
  phpmyadmin:
```

```
    image: myimagephpadmin
```

```
    build: ./phpadmin
```

```
    ports:
```

```
      - "8088:80"
```



SPERIMENTAZIONE



- AVVIO DEL «docker-compose up»

```
root@luis-Virtual-Machine:/home/luis/Scrivania/lavorotesi# docker-compose up
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm.
All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

Creating network "lavorotesi_default" with the default driver
Building web
Step 1/1 : FROM webdevops/php-apache:alpine-php7
----> 2d3f645564e9
Successfully built 2d3f645564e9
Successfully tagged myimageweb:latest
db_1      | 2019-04-08T08:26:52.762988Z 0 [System] [MY-010931] [Server] /usr
/sbin/mysqld: ready for connections. Version: '8.0.15' socket: '/var/run/mysqld
/mysqld.sock' port: 3306 MySQL Community Server - GPL.
db_1      | 2019-04-08T08:26:53.156769Z 0 [System] [MY-011323] [Server] X PL
ugin ready for connections. Socket: '/var/run/mysqld/mysqlx.sock' bind-address:
':::' port: 33060
```

NODO MANAGER

WEB SERVER: «localhost:4500»

PHPMYADMIN/MYSQL: «localhost:8088»

localhost:4500

Universita' degli studi di Bari Aldo Moro, Informatica ITPS Magno Luis Benitez Murillo



Username

Password

Login

localhost:8088/sql.php?server=

phpMyAdmin

Recent Favorites

- information_schema
- mydb
 - New
 - dispositivo
 - marca
 - rilevazione
 - tipo
 - utente

Server: db » Database: mydb » Table: utente

Browse Structure SQL S

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM 'utente'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25

+ Options

	id	password	nome
<input type="checkbox"/> Edit Copy Delete	1	641358641358	Luis
<input type="checkbox"/> Edit Copy Delete	2	641359641359	Magno
<input type="checkbox"/> Edit Copy Delete	19	rasmuslerdorf	asdas

Check all With selected: Edit Copy Delete Export

DEPLOY DELL'APPLICAZIONE SUI NODI WORKER

Viene utilizzato il comando:

«`docker stack deploy --compose-file docker-compose.yaml stackdemo`»

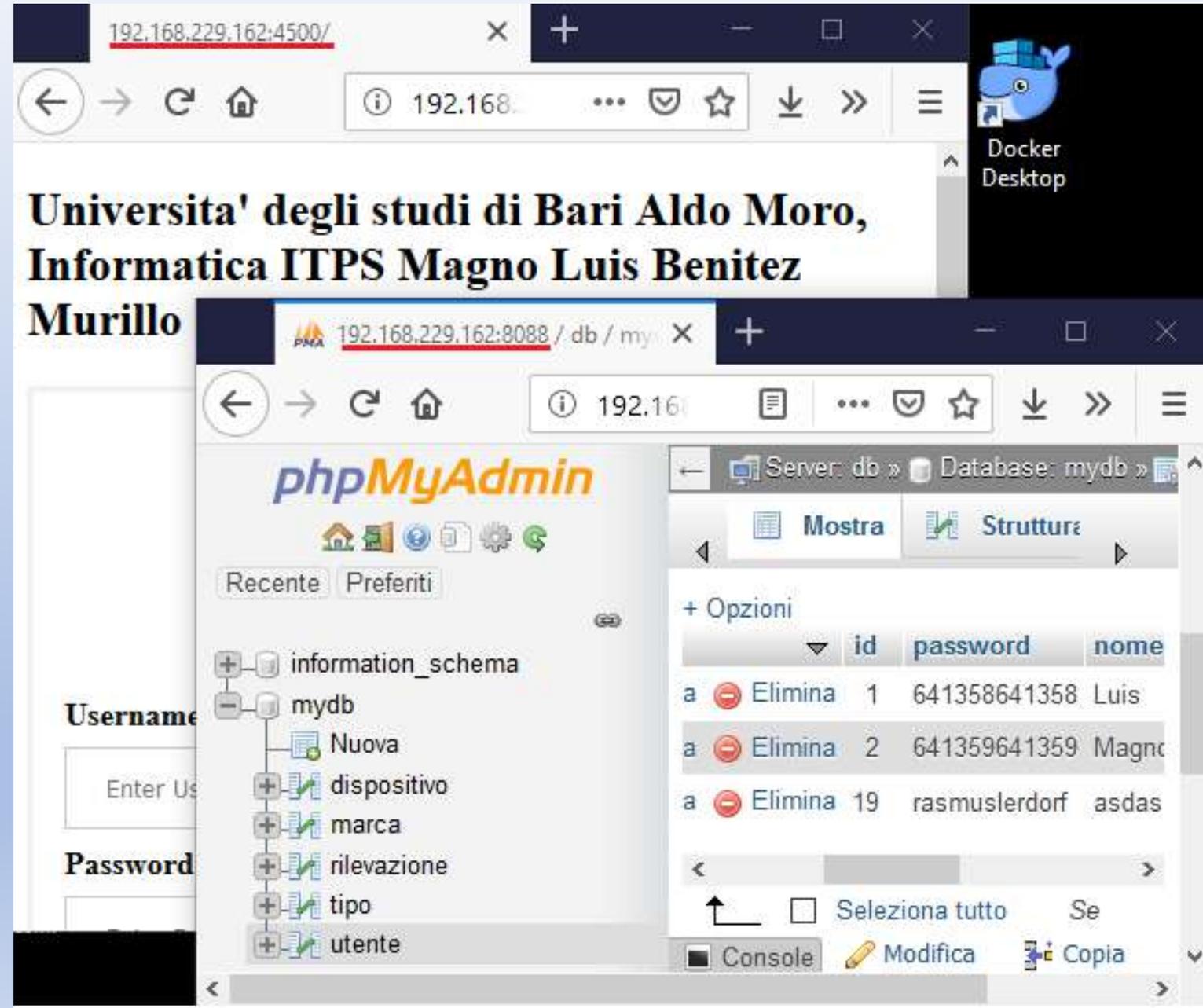
```
root@luis-Virtual-Machine:/home/luis/Scrivania/lavorotesi# docker stack deploy --compose-file  
docker-compose.yaml stackdemo  
Ignoring unsupported options: build, restart  
  
Creating network stackdemo_default  
Creating service stackdemo_phpmyadmin  
Creating service stackdemo_web  
Creating service stackdemo_db  
root@luis-Virtual-Machine:/home/luis/Scrivania/lavorotesi#
```

I nodi worker potranno utilizzare l'applicazione che il manager rilascia sullo SWARM.

WORKER

Sul nodo Worker dello Swarm:
WEB SERVER:
«192.168.229.162:4500»

Sul nodo Worker dello Swarm:
PHPMYADMIN:
«192.168.229.162:8088»



192.168.229.162:4500/

192.168.229.162:8088 / db / my

Universita' degli studi di Bari Aldo Moro,
Informatica ITPS Magno Luis Benitez
Murillo

phpMyAdmin

Server: db » Database: mydb »

Mostra Struttura

+ Opzioni

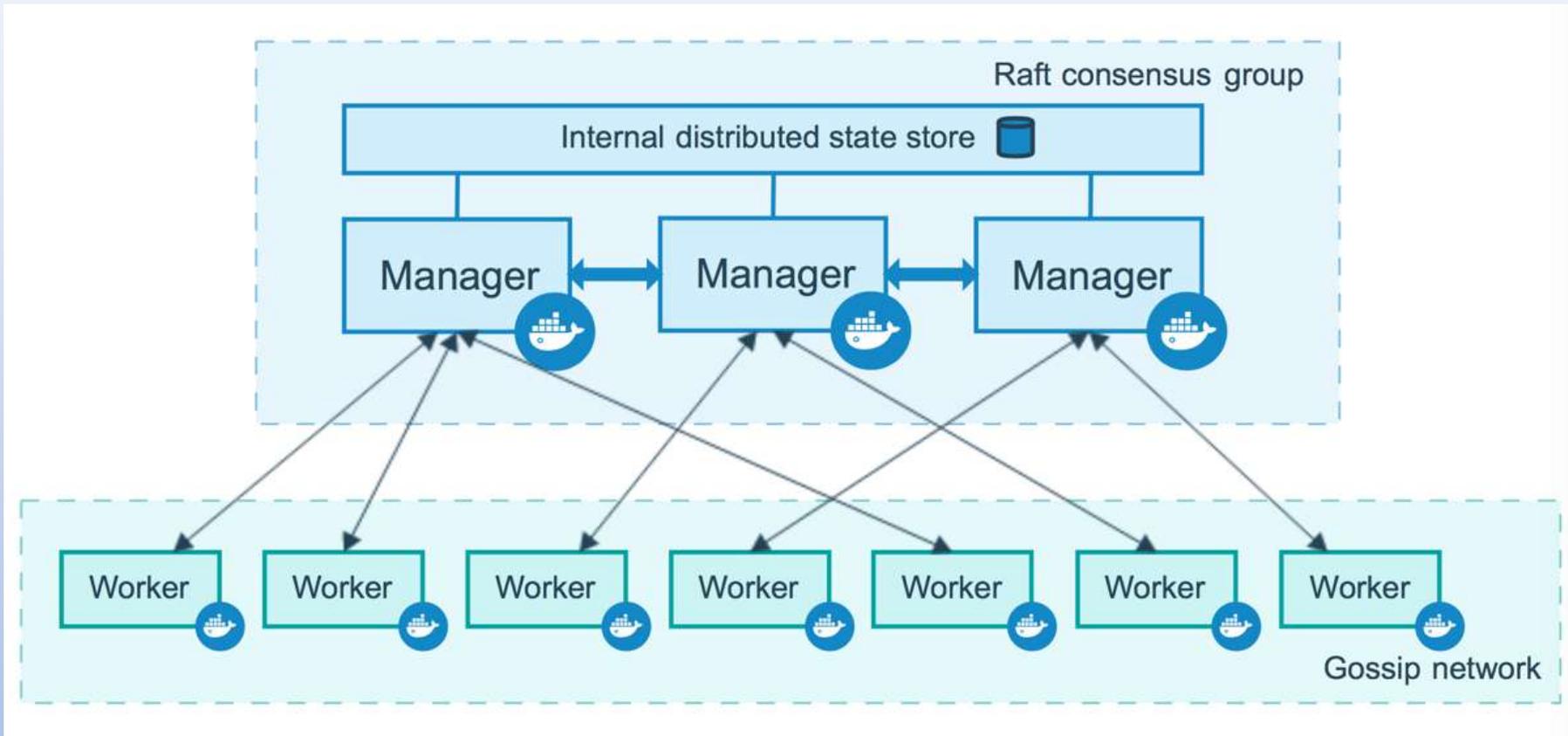
	id	password	nome
a Elimina	1	641358641358	Luis
a Elimina	2	641359641359	Magno
a Elimina	19	rasmuslerdorf	asdas

Console Modifica Copia

RISULTATI

- Senza l'uso dello Swarm si poteva optare per un vero e proprio trasferimento manuale dell'applicazione tramite i file Docker di configurazione su un'altra macchina, indipendentemente se fossero state in relazione.
- Lo Swarm invece offre un grande vantaggio in quanto tramite l'host manager basterà configurare il Docker Swarm aggiungere i nodi e poi dover rilasciare l'applicazione.
Non sarà necessario alcun trasferimento manuale di file Docker tra le macchine.

FUTURI



- Algoritmo di consenso **Raft** per gestire lo stato globale del cluster e per assicurarsi che tutti i nodi manager che sono responsabili dell'attività nel cluster stiano memorizzando lo stesso stato coerente e in caso di errore o si verificano dei problemi sul Manager leader, qualsiasi nodo Manager può raccogliere le attività e ripristinare i servizi in uno stato stabile

GRAZIE PER L'ATTENZIONE



**UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO**