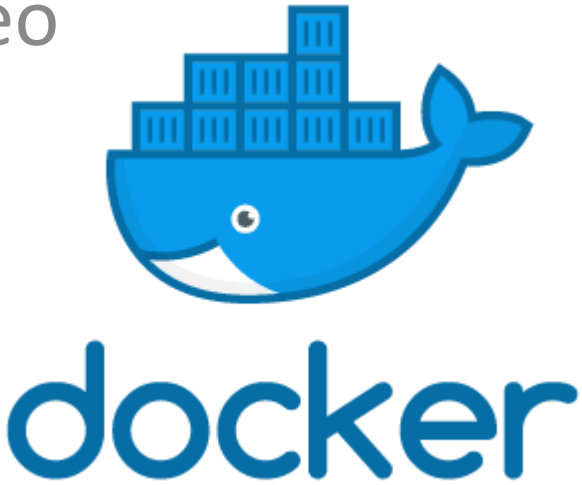
Docker

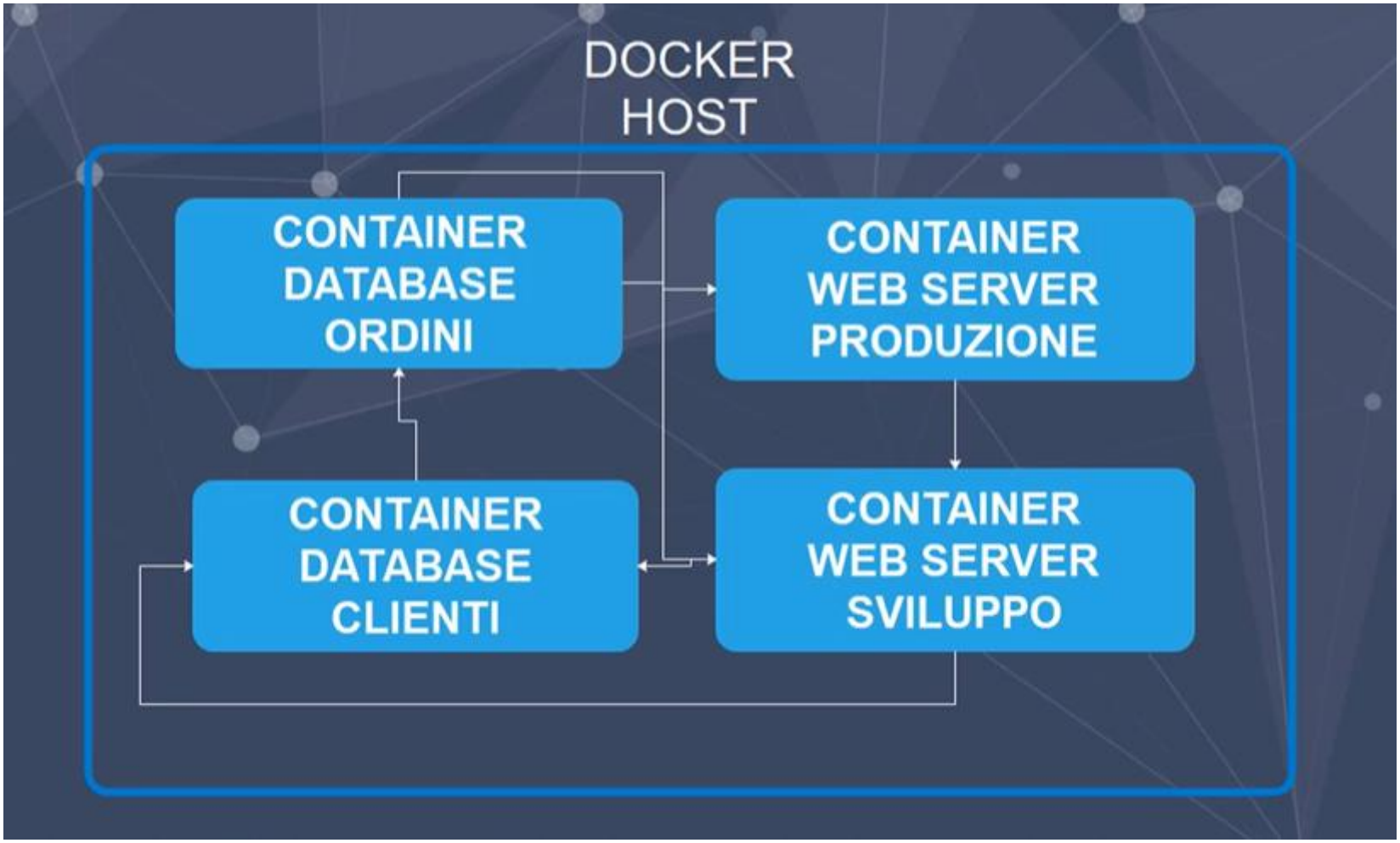
Riccardo Cattaneo



Docker Compose

Nella parte del Networking abbiamo visto come i Container possono comunicare tra di loro. Nelle architetture a Microservizi avere più container che interagiscono tra loro è lo standard.

Una situazione reale può essere rappresentata da questa immagine :


DOCKER
HOST
CONTAINER
DATABASE
ORDINI
CONTAINER
WEB SERVER
PRODUZIONE
CONTAINER
DATABASE
CLIENTI
CONTAINER
WEB SERVER
SVILUPPO

Lo Strumento Docker Compose ci permette di definire ed eseguire scenari applicativi complessi che richiedono l'utilizzo di più container.

Il principio di funzionamento di Docker Compose è il seguente :

1 – Definizione del docker-compose che è un file di testo con estensione .yml che ci permette di definire e configurare i servizi relativamente alla nostra applicazione.

2 – Esecuzione dei Servizi definiti all'interno del docker-compose

3 – Possibilità di effettuare lo «scaling» dei servizi in base al carico di lavoro e alle proprie necessità

4 – Arresto dei servizi definiti all'interno del docker-compose

Il componente «docker compose» è già compreso nell'installazione di docker nei sistemi Windows e Mac. Su linux è necessario installarlo separatamente.

```
C:\WINDOWS\system32>docker-compose version
docker-compose version 1.27.4, build 40524192
docker-py version: 4.3.1
CPython version: 3.7.4
OpenSSL version: OpenSSL 1.1.1c  28 May 2019

C:\WINDOWS\system32>
```

```
ubuntu@ip-10-0-0-41:~$ docker-compose -version

Command 'docker-compose' not found, but can be installed with:

sudo snap install docker          # version 18.06.1-ce, or
sudo apt  install docker-compose

See 'snap info docker' for additional versions.

ubuntu@ip-10-0-0-41:~$ clear
```

Cos’è il formato YAML

YAML è un formato per la serializzazione dei dati facilmente leggibile dagli esseri umani. Sviluppato nel 2001, inizialmente YAML era l’acronimo di “Yet Another Markup Language” (Ancora un altro linguaggio di markup) che è andato perso in favore di “YAML Ain’t a Markup Language” (YAML Non è un linguaggio di markup) per richiamare la memorizzazione dei dati del linguaggio, al contrario dei soliti linguaggi di markup.

YAML prende ispirazione da molti linguaggi di programmazione ed è disponibile per molti di essi come Java, C/C++, Python e Javascript.

Per alcuni sviluppatori YAML è simile al JSON, entrambi sono dei formati per scambiare dati ma la principale differenza tra i due è che la priorità principale di JSON è la semplicità e l'universalità del formato.

Garantire questi obiettivi compromette la leggibilità dei dati da parte degli essere umani, per questo motivo la priorità assoluta per **YAML è di fornire un formato dati facilmente leggibile dagli esseri umani** e offrire un supporto per la serializzazione di strutture di dati native. **Un file YAML è più difficile da generare e analizzare, ma è più facile da leggere**, ed offre alcuni tipi di dato da utilizzare per il passaggio da un linguaggio di programmazione ad un altro, ciò comporta che un file JSON è anche un file YAML perfettamente valido.

Sintassi YAML

Ecco alcune regole di sintassi per scrivere un file YAML:

- Vengono utilizzati gli spazi per denotare le strutture;
- Il commenti iniziano con il simbolo del cancelletto (#);
- Le liste sono denotate con il simbolo del trattino (–);
- Gli array associativi sono rappresentati utilizzano il simbolo dei due punti (:).

Struttura di Docker Compose

Il file «docker-compose.yml» si definisce utilizzando il formato YAML. Come abbiamo detto è necessario definire quali saranno i servizi da eseguire e quali caratteristiche questi dovranno avere. I comandi principali sono :

docker-compose up **(avvio)**

docker-compose down **(stop)**

Per prima cosa dobbiamo creare una cartella ad esempio la chiamiamo «composefile» e dentro andiamo a creare il file «docker-compose.yml». La prima cosa che si scrive in un file compose è la versione in questo modo :

version: ‘3’

La seconda cosa da scrivere sono i servizi in questo modo :

```
services:
        web:
                image: nginx

        database:
                image: redis
```

IMPORTANTE : non usare il tasto «tab» per l'indentazione del codice nei file yml, provoca errore.

Una volta salvato il file possiamo verificare che abbiamo scritto tutto correttamente attraverso il comando **docker-compose config**.

```
riccardo@riccardo-VirtualBox:~/composeFile$ docker-compose config
services:
  database:
    image: redis
  web:
    image: nginx
version: '3.0'
```

Per eseguire il file scriviamo

docker-compose up –d

Per verificare possiamo lanciare docker image ls e poi docker ps

docker-compose down

Per approfondire sui comandi possibili da utilizzare si rimanda alla documentazione ufficiale.

https://docs.docker.com/compose/

Opzione Ports

Facciamo un piccolo passo avanti per quanto riguarda il docker compose, andando a vedere come si possono configurare le porte attraverso l'opzione **ports**.

```
version: '3'

services:

 web:
  image: nginx
  ports:
   - 8090:80/tcp
 database:
  image: redis
```

Opzione Scale

Con docker compose abbiamo la possibilità di effettuare lo «scaling» dei servizi in base al carico di lavoro. Riprendendo l'esempio precedente, con i due container web e database. Supponiamo che il nostro carico di lavoro stia aumentando relativamente alla parte dei database, e che quindi vogliamo scalare il nostro database.

docker-compose up –d --scale database=10

docker ps

docker-compose down