**Microservizi:**

componenti indipendenti tra loro, ognuno con i propri dati. **Piccole parti di un’applicazione.**

Es microservizi: “aggiungi al carrello” e funzione “cerca” o autenticazione in un e-commerce.

* Microservizi sono **indipendenti tra loro**
* Si occupano di **specifiche funzionalità**
* **Comunicano** tra loro

Vantaggi:

: facilità di sviluppo

: facilità di manutenzione

* **Docker: container di microservizi**
  + Composto da:
    - Client: interazione con l’utente
    - Server: processo in background che gestisce il container
    - Registry: sistema dove vengono salvate le immagini da eseguire (simile a un database)
  + Controlli:
    - help: mostra informazioni su comandi
    - build: creazione
    - run: avvia l’applicazione
    - push: immagine inviata
    - pull: immagine scaricata
    - images: elenca tutte le immagini
    - ps: elenca tutti i container
    - stop: Ferma un container
    - rm: rimuove un container
    - rmi: rimuove un’immagine
    - exec: esegue un comando di un container
    - logs: visualizza i log. Servono a tracciare le attività dei container. Mostra se un log è o meno in esecuzione
    - curl: permette di effettuare chiamate al server
* **Container docker**: simile a una JVM, fornisce un ambiente di esecuzione. **Include l’applicazione e le dipendenze**
* **Immagine**: file con **istruzioni per creare un container**. da un’immagine si possono creare più container. Le immagini sono mantenute in repository (come Docker Hub)
* **Root**: **directory,** la cartella principale contenente tutti i file e le sottocartelle
* **Logs:** registrazioni di eventi

Creazione di un progetto: SpringBoot, dipendenze “Spring Web”

Commenti:

* FROM: usa l’immagine base di open JDK 11
* WORKDIR /app: imposta la directory
* COPY: copia il .jar dall’host al container
* EXPOSE 8080: espone la porta (8080) sulla quale l’applicazione sarà in ascolto
* ENTRYPOINT: specifica il comando per eseguire l’app

Il Docker deve essere posizionato nella root del progetto e del pom.xml

Mappatura porte: i container non sono raggiungibili dal’esterno per default

>> docker run -p 8080:8080 docker-java-demo -> mappa la porta 8080 dell’host tale che sia raggiungibile dal client

Chiamate sincrone: gestite una alla volta, (un comando alla volta, se questo fallisce si blocca tutto)