….

I docker file sono file di istruzioni che indicano a Docker come deve generare l’immagine di un container.  
Sono file testuali.  
Devono iniziare con una **FROM** instruction (indica l’immagine base da cui partire).  
Es

**FROM** openjdk:8-jdk-alpine   
**MAINTAINER** baeldung.com   
**COPY** target/docker-message-server-1.0.0.jar message-server-1.0.0.jar **ENTRYPOINT** ["java","-jar","/message-server-1.0.0.jar"]

Text

Description automatically generated

Quando si ha il docker file, si usa **docker build** per creare un immagine docker dal docker file.  
A quel punto si puo’ lanciare **docker run** per far startare il container.  
(una build o la si pusha su un registro online, o si lascia locale, in entrambi i casi la run la riconosce usando i tag)

Nello specifico il giro per applicazioni spring boot, è che si crea il docker file in cui si indica che il jar (precedentemente ottenuto) dell’applicazione deve essere copiato nell’immagine, e che sarà l’entry point (cioè che viene lanciato il .jar quando parte il container).  
In questo modo quando si lancia il container con il docker run parte l’applicativo spring boot.

Spring Boot da 2.3 evita di dover creare il dockerfile per arrivare ad avere un’immagine startabile su docker della nostra applicazione.  
Basta fare **mvn spring-boot:build-image**

Ma il processo docker file -> docker build -> docker run, è tedioso visto che va fatto per ciascun applicativo / container.  
Per questo esiste **docker-compose che tramite un unico file, docker-compose.yml** permette di configurare differenti container, tutti nello stesso file, e startarli tutti assieme.  
  
**docker-compose up –-build**

**Docker-compose down**