

# RabbitMQ

Link alla pagina docker hub: [https://hub.docker.com/\\_/rabbitmq/](https://hub.docker.com/_/rabbitmq/)

Scaricare l'immagine

```
docker pull rabbitmq
```

Eeguire il daemon

```
docker run -d --hostname my-rabbit --name some-rabbit -p
15672:15672 -p 5672:5672 -v
/home/silvia/ProgettoDist/rabbitmq:/var/lib/rabbitmq
rabbitmq:3-management
```

Questo avvierà rabbitMQ: sulla porta 5672 ci saranno le API, all'indirizzo *127.0.0.1:15672* ci sarà l'interfaccia di management (username e password: guest / guest)

-v /home/silvia/ProgettoDist/rabbitmq:/var/lib/rabbitmq è il volume: tutti i file prodotti saranno disponibili anche dopo che il container verrà chiuso al percorso locale specificato nella parte sinistra dei due punti (cambiare secondo le proprie necessità).

Con i tutorial ero rimasta qui: <https://www.rabbitmq.com/tutorials/tutorial-two-java.html>

Altri link:

<https://github.com/rabbitmq/rabbitmq-tutorials/tree/master/java>

---

# MySQL

Link alla pagina docker hub: [https://hub.docker.com/\\_/mysql/](https://hub.docker.com/_/mysql/)

Scaricare l'immagine

```
docker pull mysql
```

Per avviare un solo container:

```
docker run --name <nome_container> -v PATH_LOCALE:/var/lib/mysql
-e MYSQL_ROOT_PASSWORD=passRoot -p 3306:3306 -d mysql:5.7
```

**3306:3306** -> a sinistra c'è la porta che si andrà a utilizzare, a destra c'è la porta di default di MySQL che utilizza il container.

Basta utilizzare 5 numeri di porta diversi per avere 5 repliche diverse (credo).

**<nome\_container>** -> nome a scelta per il container (es: replica1, replica2, ...)

**PATH\_LOCALE** -> path locale per il salvataggio dei file del db (es: 5 cartelle)

Però non è così che dovremo avviare il tutto!

**Docker compose** è un tool che permette di avviare più container contemporaneamente (però devo capire bene come fare...).

Link utili:

<https://www.linux.com/learn/introduction-docker-compose-tool-multi-container-applications>

<https://github.com/docker/compose>