# Docker Datamint Project

Docker is a new technology that exploit the idea of container. Ideal for enabling microservices application development. Accelerate development, deployment and rollback of tens or hundreds of containers composed as a single application.

## What have been done

Everything has been done is deployed on the following github space: [github](https://github.com/havanaIt/datamint)

* Dockerfile of Zookeeper, Datamint and Kafka have been created and pushed on the above github link
* dockercompose.yml to encapsulate all the previous Dockerfile pushed on the same link
* created three Virtual Machine (VM) and set up a swarm (Manager-Worker1-Worker2)

## To Do

* Try to run Datamint docker package on the machines of the docker swarm.
* Produce and consume on such a cluster.

## How to create Datamint brick

Datamint is a suite of products act to make big data exploitable by anyone. Before deploying such a solution on a cluster the packaging of the Datamint suite must be done. The tool [docker-compose build](https://docs.docker.com/compose/reference/build/) allows to achieve such a goal. In particular, the interactions between the services: Zookeeper, Kafka and Datamint must be described in the a specific file: *docker-compose.yml* (stored on [github](https://github.com/havanaIt/datamint) repository). The instance of the three services is then achieved by typing:

* docker-compose build PATH/TO/docker-compose.yml

## How to create a Docker Swarm

Swarm is a master/slave node system. A manager node is chosen by the user between all the swarm nodes. The other are worker or could be manager in order to guarantee the redundancy.

Managers elect a leader that handles the load balancing of the applications, check the heart beating of the workers and decides the different node worker activity.

* docker-machine swarm init –advertise-addr <MANAGER IP> (define the manager)
* docker-machine ssh <MANAGER name> (ssh on the manager node)
* Once on the manager node:
  + docker swarm join-token worker (returns a token to add worker to the swarm)
  + docker swarm join-token manager (returns a token to add manager to the swarm)

## How to deploy applications on swarm cluster

Deployment on a swarm cluster is obtained by create a sort of application images repository that is pulled from the manager to the worker nodes.

Below the steps to deploy applications (services) on the worker swarm nodes:

* create an images repository where images are pushed on (see [docker registry](https://docs.docker.com/registry/#what-it-is))
* personalize your docker-compose.yml with decoration:
  + [node==worker] or [node==manager] if you want to install an application just on a specific worker/manager (load balancing)[[1]](#footnote-1)
* deploy images from the manager node running:
  + docker stack deploy –c docker-compose.yml <.>[[2]](#footnote-2) [(stack deploy)](https://docs.docker.com/get-started/part5/#persist-the-data)

1. If decoration are not use then the images are, depending on the replication factor, installed in a round robin scheduling by default [↑](#footnote-ref-1)
2. <.> chose a name [↑](#footnote-ref-2)