Ambiente

# Sistema Operativo Macchina Host

Linux Kubuntu 16.04

Linux kernel version 4.4.0-31-generic

# Docker Client/ Server

Client:

Version: 1.13.0

API version: 1.25

Go version: go1.7.3

Git commit: 49bf474

Built: Tue Jan 17 09:58:26 2017

OS/Arch: linux/amd64

Server:

Version: 1.13.0

API version: 1.25 (minimum version 1.12)

Go version: go1.7.3

Git commit: 49bf474

Built: Tue Jan 17 09:58:26 2017

OS/Arch: linux/amd64

Experimental: false

Docker Machine : docker-machine version 0.9.0, build 15fd4c7

Docker Compose: docker-compose version 1.10.0, build 4bd6f1

Virtual Box: 5.0.32

# Sviluppo

Eclipse version: Neon.2 Release (4.6.2)

Maven: Apache Maven 3.3.9

Git version: 2.7.4

Nodejs version: 4.2.6

NPM version: 3.5.2

Java version: openjdk 1.8.0\_121

Guida

## Docker

Installazione Docker run:

$ sudo apt-get **update**  
  
$ sudo apt-**get** **install** curl linux-image-extra-$(uname -r) linux-image-extra-**virtual**

**$ sudo apt-get install apt-transport-https ca-certificates**

**$ curl -fsSL https://yum.dockerproject.org/gpg | sudo apt-key add -**

**$ apt-key fingerprint 58118E89F3A912897C070ADBF76221572C52609D**

**$ sudo apt-get install software-properties-common  
$ sudo add-apt-repository \  
 "deb https://apt.dockerproject.org/repo/ \  
 ubuntu-$(lsb\_release -cs) \  
 main"**

**$ sudo apt-get update**

**$ sudo apt-get -y install docker-engine**

**$ apt-cache madison docker-engine**

**Choose a specific version to install. The second column is the <version\_string>**

**In this project i used 1.13.0-0**

**$ sudo apt-get -y install docker-engine=<VERSION\_STRING>**

**Manage Docker as a non-root user**

**$ sudo groupadd docker**

**$ sudo usermod -aG docker $USER**

**May need a Log Off**

**Configure Docker to start on boot**

**$ sudo systemctl enable docker**

## Docker-Machine

**$ curl -L https://github.com/docker/machine/releases/download/v0.9.0/docker-machine-`uname -s`-`uname -m` >/tmp/docker-machine &&  
 chmod +x /tmp/docker-machine &&  
 sudo cp /tmp/docker-machine /usr/local/bin/docker-machine**

**$ docker-machine version**

**Queste due operazioni non sono fondamentali da eseguire nel caso in cui non dovesse funzionare il case study**

**//$ docker-machine create *--driver virtualbox default***

***//$ eval "$(docker-machine env default)"***

## Docker-compose

**$ curl -L "https://github.com/docker/compose/releases/download/1.10.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose**

**$ chmod +x /usr/local/bin/docker-compose**

**To check**

**$ docker-compose *--version***

## AcmeAIR

Our Case study will be AcmeAir a Node.js implementation of the Acme Air Sample Application. With datastore support of MongoDB, Cloudant, Cassandra.

With runtime support of Docker in order to have a Micro-Services Architecture to Analyze

## Building and Start

**$ git clone** [**https://github.com/wasperf/acmeair-nodejs.git**](https://github.com/wasperf/acmeair-nodejs.git)

**$ cd acmeair-nodejs**

**$ npm install**

**$ cd node\_modules/.bin**

**$ npm install**

**$ cd ..**

**$ docker network create --driver bridge my-net**

**$ set NETWORK=my-net && export NETWORK=my-net**

**The Dockerfile build may take a while**

**$ docker-compose build**

**The “up” command start the service in only one windows, in the terminal you will see some log of the startup of the application and some logs of communications**

**$ docker-compose up**

**After 1~2 min all the services should be ready (the first startup take longer)**

**$ Go to http:://127.0.0.1:80/main/acmeair**

**If this page doesn’t appear wait a bit more or come back to Docker-machine guide e type the two missing commands.**

Now you can use the system, the Support Service is not working.

## Eclipse and run app

Download Eclipse.

Define a workspace in this example in ~/EclipseWorkspace

Run it.

The Application is in a Github. Clone the repository and copy inside the Eclipse workspace

**$ git clone** [**https://github.com/kansor/Univaq-Thesis-GM.git**](https://github.com/kansor/Univaq-Thesis-GM.git)

**$ cd Univaq-Thesis-GM/ArchitectureImpactAnalysis/**

**$ cp ArchitectureRecovery ~/EclipseWorkspace**

Inside Eclipse go in the Project Explorer.

**$ Right-Click**

**$ Import**

**$ Existing Maven Project**

**$ Select ~/EclipseWorkspace**

**$ Finish**

**Wait until the maven building ends.**

## 