Fiware installation

Prerequisites :

DOCKER AND DOCKER COMPOSE

To keep things simple all components will be run using **Docker**. **Docker** is a container technology which allows to different components isolated into their respective environments.

We used it on “Linux” environment.

- First of all, we have to install Docker on Linux. follow the instructions here :

Before you install Docker CE for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

SET UP THE REPOSITORY

1. Update the apt package index:

```
$ sudo apt-get update
```

2. Install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get install \
apt-transport-https \
ca-certificates \
curl \
software-properties-common
```

3. Add Docker’s official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Verify that you now have the key with the fingerprint9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, by searching for the last 8 characters of the fingerprint.

```
$ sudo apt-key fingerprint 0EBFCD88


pub   4096R/0EBFCD88 2017-02-22


      Key fingerprint = 9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid                  Docker Release (CE deb) <docker@docker.com>
```

```
sub   4096R/F273FCD8 2017-02-22
```

4. Use the following command to set up the **stable** repository. You always need the **stable** repository, even if you want to install builds from the **edge** or **test** repositories as well. To add the **edge** or **test** repository, add the word edge or test (or both) after the word stable in the commands below.

   Note: The lsb_release -cs sub-command below returns the name of your Ubuntu distribution, such as xenial. Sometimes, in a distribution like Linux Mint, you might need to change $(lsb_release -cs) to your parent Ubuntu distribution. For example, if you are using Linux Mint Rafaela, you could use trusty.

   - x86_64 / amd64
   - armhf
   - IBM Power (ppc64le)
   - IBM Z (s390x)

```
$ sudo add-apt-repository \
  "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) \
  stable"
```

   Note: Starting with Docker 17.06, stable releases are also pushed to the **edge** and **test**repositories.

INSTALL DOCKER CE

1. Update the apt package index.

```
$ sudo apt-get update
```

2. Install the *latest version* of Docker CE, or go to the next step to install a specific version:

```
$ sudo apt-get install docker-ce
```

   Got multiple Docker repositories?

   If you have multiple Docker repositories enabled, installing or updating without specifying a version in the apt-get install or apt-get update command always installs the highest possible version, which may not be appropriate for your stability needs.

3. To install a *specific version* of Docker CE, list the available versions in the repo, then select and install:

   a. List the versions available in your repo:

```
$ apt-cache madison docker-ce
```

```
docker-ce | 18.03.0~ce-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages
```

b. Install a specific version by its fully qualified package name, which is package name (docker-ce) "=" version string (2nd column), for example, docker-ce=18.03.0~ce-0~ubuntu.

```
$ sudo apt-get install docker-ce=<VERSION>
```

The Docker daemon starts automatically.

4. Verify that Docker CE is installed correctly by running the hello-world image.

```
$ sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

Docker CE is installed and running. The docker group is created but no users are added to it. You need to use sudo to run Docker commands.

Fiware installation :

It is to note that Fiware is an existant platform and we are not the ones that developed it. Fiware is known by a unique and complex architecture that is based on its own components. We have tried different approached in Fiware's installation and decided to use a project that has been worked on by Fiware's partner ATOS. The installation and configuration steps are presented below:

1. Clone the repository with the following command:

```
https://github.com/Atos-Research-and-Innovation/IoTagent-LoRaWAN.git
```

2. Once the repository is cloned, you have to download the dependencies for the project. To do so, from the root folder of the project execute:

```
npm install
```

3. If you ever want to test the IoT Agent or make sure that it runs correctly, you can run it with the default configuration by executing the following command

```
node bin/iotagent-lora
```

The bootstrap process should finish with:

```
info: Loading devices from registry
info: LoRaWAN IoT Agent started
```

4. Check that the IoTA is running correctly:

```
curl -v http://localhost:4061/iot/about
```

The result must be similar to:

```
{"libVersion":"2.6.0-next","port":4061,"baseRoot":"/"}
```

5. To run orion manually for test purposes, you can run the following commands

   - Manually, run MongoDB on another container for the data to be stored in it. Then, run orion while linking it to the already running MongoDB docker:

     sudo docker run --name mongodb -d mongo:3.2

     sudo docker build -t orion .

     sudo docker run -d --name orion1 --link mongodb:mongodb -p 1026:1026 orion -dbhost mongodb.

   - Specify where to find your MongoDB host:

     sudo docker build -t orion .

     sudo docker run -d --name orion1 -p 1026:1026 orion -dbhost <MongoDB Host>.

Check that everything is correctly working with the command below

    curl localhost:1026/version

PS: The parameter -t orion in the docker build command gives the image a name.

Docker-compose

For simplicity reasons, you can also run all the dockers in one go by using docker-compose. You can configure as many containers as you want, how they should be built and connected, and where data should be stored.

A docker-compose.yml file

A docker-compose.yml file is a YAML file that defines how Docker containers should behave in production.

docker-compose.yml

You can run a single command to build, run, and configure all of the containers.This will be done by running the following command:

```
docker-compose -f docker/docker-compose.yml up
```

The following figure presents the docker-compose.yml file with which we can run all the dockers containing fiware's components with a single command:

```
version: '2'

volumes:
  mongo_data:

services:

  mongodb:
    expose:
      - "27017"
    hostname: mongodb
    image: mongo:3.2
    stdin_open: true
    tty: true
    volumes:
      - mongo_data:/data/db

  orion:
    command:  -dbhost mongodb -port 1026 -logLevel DEBUG
    depends_on:
      - mongodb
    expose:
        - "1026"
    hostname: orion
    image: fiware/orion:latest
    ports:
      - "1026:1026"
    stdin_open: true
    tty: true

    iotagent-lora:
      depends_on:
        - mongodb
        - orion
      entrypoint: bin/iotagent-lora docker/config-docker.js
      hostname: iotagent-lora
      image: ioeari/iotagent-lora
      ports:
        - "4061:4061"
      stdin_open: true
      tty: true
```

This docker-compose.yml file tells Docker to do the following:

- Pull the image mongodb from mongo 3.2, the image orion from fiware/orion:latest and the image iotagent-lora from ioeari/iotagent-lora.
- Immediately restart containers if one fails.
- Map port 1026 on the host to web's port 1026.
- Map port 4061 on the host to web's port 4061.