FM SETUP INSTRUCTIONS

Index

System requirements

1. Install docker(https://docs.docker.com/engine/install/)

2. Install docker-compose(https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-compose-on-ubuntu-20-04)

3. docker-compose version needs to be greater than 1.29.2

4. Ideal configuration :

```
RAM : 4GB
CORES: 4
```

4. Install jq

```
apt-get install jq
```

Setup FM on localhost or through ssh access

You would need access to fleet_manager repository to do this

1. Clone fleet_manager repository
2. Build fleet_manager docker images

Setup FM on air-gapped servers

1. Built docker images would be available in data server (data@192.168.10.21), at the location : /atidata/datasets/fm_setup_v<fm_version>

2. Copy the entire fm_setup_v<fm_version> folder to the respective air-gapped server

3. ssh to the remote server, do the following - This would build the required docker images on the air-gapped server

```
cd fm_setup_v<fm_version>
bash load_images.sh
```

4. Create a directory (static dir)

```
mkdir static
```

5. Copy docker_compose_v<fm_version>.yml file to the static folder you created

6. With this FM installation is done

Start using fleet_manager

1. Go to the static directory where docker_compose_v<fm_version>.yml was copied

2. Run the following command to get all the ips of the server

```
ifconfig | grep inet | awk '{print $2}' | egrep '[0-9]+\.[0-9]+\.[0-9]+\.
[0-9]+'
```

3. Run the following command

```
docker-compose -p fm -f docker_compose_v<fm_version>.yml up
```

4. Create ssl certs (This has to be done only for the first time). ip_1, ip_2...ip_n etc are ips of the FM server which were obtained in step 2

```
docker exec -it fleet_manager bash
create_certs "127.0.0.1,<ip_1>,<ip2>,...,<ip_n>"
```

4. Post creating the certs, exit the docker container and Restart FM

5. You should be able to access fm in the url <https://<fm_ip>/fm>

Clone fleet_manager repository

1. Clone fm repo

```
git clone https://<git_token>@github.com/AtiMotors/fleet_manager
cd fleet_manager
git pull
git submodule init
```

```
```

2. Change mule submodule branch to dev

   a. open and edit .git/config file, add branch=dev to submodule mule entry

   ```
   [submodule "mule"]
   url = https://<git_token>@github.com/AtiMotors/mule.git
   active = true
   branch = dev
   ```

3. Update submodules

```
git submodule update --recursive
```

Build fleet_manager docker images

1. From the cloned fleet manager repo, run setup_fm script

```
./scripts/setup_fm.sh
```

2. There will be prompts to help you build images.

```
prompts:
a. Want to build images on a remote server? (y/n) - choose based on whether
you want build fm on localhost or remote server through ssh

b. Should build base images? (y/n) - Base images will have to be installed
if Dockerfile.base, pyproject.toml has been modified in any of the repos
(fleet_manager, fm_plugins). You might also have to do it if you are build
fm for the first time. It is better to build base images in system that
have more than 2 cores. This would take about half an hour

c. Enter static data folder path in the remote server (~/static) : You will
have enter the data path in remote server. To this location on the docker-
compose file would be copied.
```

3. docker_compose_v<fm_version>.yml would be created, copied to fleet_manager/static dir if you built fm on localhost else you find the docker-compose file in inputted static data path in the remote server

4. With this FM installation is done.

Restart FM

1. Run the following command from the static directory

```
docker-compose -p fm -f docker_compose_v<fm_version> down
docker-compose -p fm -f docker_compose_v<fm_version> up
```

Run CI

1. We have created a script ci.sh, which can be used to build and push images to sanjaya.atimotors.com

2. Arguments required to run the script.

```
cd <fleet_manager repo>
bash scripts/ci.sh <y/n> <sanjaya_login_username> <sanjaya_login_password>

###
arg 1 - y/n - If 'y' is given ci script would assume it is production build
else 'n' is inputted the build would be considered non-prod

arg 2 - Sanjaya login user name

arg 3 - Sanjaya login password
####
```

3. What does the script do?

   a. Builds all the required docker images. The image to be built is obtained from misc/docker-compose_untagged.yml file**

   b. The docker images would be tagged with the checked out version or tag

   c. Creates a file static/docker_compose_v<branch/tag>.yml

   d. Uploads docker_compose_v<branch/tag>.yml to sanjaya.atimotors.com - The file should get uploaded to the path: static_master_fm/downloads/<prod/non-prod>/fm/fm_v<branch/tag>

   e. Uploads a file release.dt (When was the last commit, when images were created) to sanjaya.atimotors.com - The file should get uploaded to the path: static_master_fm/downloads/<prod/non-prod>/fm/fm_v<branch/tag>

   e. Pushes the images created to registry in sanjaya.atimotors.com