FM Support

Index

Setup sherpas

1. Copy fm cert file(fm_rev_proxy_cert.pem) from <static_dir>/certs to sherpa's /opt/ati/config directory

2. Add this patch to /opt/ati/config/config.toml in the mule

```
[fleet]
api_key = <api_key>
chassis_number = <chassis_number>
ip="<fm_ip_address>"
port="443"
fm_cert_file="/app/config/fm_rev_proxy_cert.pem"
```

Access Config Editor

1. The config editor should be accessible at <https://<fm_ip>/config_editor>
2. Credentials for login in to config editor can be obtained from docker-compose_v<fm_version>.yml file (would be available in the static folder on the FM server).

```
ME_CONFIG_BASICAUTH_USERNAME: ****
ME_CONFIG_BASICAUTH_PASSWORD: ****
```

Send updates to master fm

1. Check whether FM server has access to sanjaya.atimotors.com by doing a ping

```
ping sanjaya.atimotors.com
```

2. Use config editor, select the database fm_config, select the collection master_fm, click on the document to edit it
3. If sanajaya.atimotors.com is accessible, you don't have to follow the remaining steps(steps 4-6) a. Change the below mentioned parameters in the document, save the same. And Restart FM. The remaining steps need not be followed

   ```
   api_key: '<api_key generated for the customer>'
   send_updates: true
   ```

4. If sanajaya.atimotors.com is not accessible but you are be able to ssh to FM server via another machine which has access to sanjaya.atimotors.com, then a reverse tunnel can be setup to access sanjaya.atimotors.com
5. To setup reverse tunnel, get mfm_rev_tunnel.tar from the downloads section on the dashboard and copy the same to the machine which has access sanjaya.atimotors.com(pingable) and has ssh access to the FM server, do the following

```
tar -xvf mfm_rev_tunnel.tar ## This is for Linux, something similar has to
be done for other os
cd mfm_rev_tunnel
bash mfm_rev_tunnel.sh <user@fm_server_ip> <client_name>
```

6. ssh into the FM server, set GatewayPorts to yes in /etc/ssh/sshd_config (This will require sudo access) and restart the ssh service, reboot the machine for the this to take effect.

```
sudo systemctl restart ssh
```

6. Follow step 2, Change the below mentioned parameters in the document, save the same

```
mfm_ip: '<fm_server_ip>'
mfm_ip: '9010'
http_scheme: 'http'
ws_scheme: 'ws'
api_key: '<api_key generated for the customer>'
send_updates: true
```

7. Restart FM

Setup auto parking feature

1. Use the config editor, select the database fm_config, select the collection conditional_trips, click on the document to edit it
2. Edit auto park params, save the same

```
auto_park: {
    book: true,
    threshold: 600, ### Threshold in seconds after which sherpa should be
sent to parking station if ## found idle
    priority: 1 ## trip priority to be given to auto park trips
}
```

3. Restart FM

Setup battery swap trips

1. Use the config editor, select the database fm_config, select the collection conditional_trips, click on the document to edit it
2. Edit battery_swap params, save the same

```
battery_swap: {
    book: true,
    max_trips: 2, ### max number sherpas that can do battery swap trips
simultaneously
    threshold: 15, ### Threshold battery level
    priority: 10 ### trip priority to be given to battery_swap trips
}
```

3. Restart FM

Setup optimal dispatch config

1. Optimal dispatch logic tries to allocate the pending trips with the best sherpa available. Choice of best sherpa is made with the paramter Z

$Z=(eta)^a/(priority)^b$
$priority=p1/p2$

```
where,
    eta - expected time of arrival computed for the sherpa to reach the
first station of the trip booked,
    priority - measure of how long a trip has been pending,
    p1 - Time since booking of currrent trip,
    p2 - Minimum of time since booking across all the pending trips,
    a - eta power factor , 0<a<1,
    b - priority power factor , 0<b<1,
```

2. Use the config editor, select the database fm_config, select the collection optimal_dispatch, click on the document to edit it

3. **Maximise number of trips done**: To get maximum number of trips done in a given time frame eta_power_factor can be set to 1, priority_power_factor can be set to 0. This will make the optimal disaptch logic to lean towards trips that can be started faster. The trip booking order will not be followed.

```
method: 'hungarian',
prioritise_waiting_stations: true,
eta_power_factor: 0.9999 ## 0.00001-0.99999,
priority_power_factor: 0.0001 ## 0.00001-0.99999,
max_trips_to_consider: 5,
```

4. **Fair scheduling**: To configure optimal dispatch logic to take trips in the order they were booked eta_power_factor can be set to 0, priority_power_factor can be set to 1.

```
method: 'hungarian',
prioritise_waiting_stations: true,
eta_power_factor: 0.0001 ## 0.00001-0.99999,
priority_power_factor: 0.9999 ## 0.00001-0.99999,
max_trips_to_consider: 5,
```

5. **Custom configuration**: There is no ideal combination of eta_power_factor, priority_power_factor. They should be choosen according to the frequecy of trip bookings, route length between the stations to maximise the throughtput.

6. For good takt time, eta power factor should be higher, for fair scheduling priority power factor should be set higher.

7. To reduce computation load due to optimal dispatch, max_trips_to_consider has been set to 5. Optimal dispatch logic will be consider only the first <max_trips_to_consider> number of trips. Default is set to 5. This can be increaded to <number_of_sherpas per fleet> in case there are more than 5 sherpas

```
[optimal_dispatch]
max_trips_to_consider=<number_of_sherpas per fleet>
```

Generate api keys

1. To generate api key with hardware id (sherpa or other smart devices)

```
docker exec -it fleet_manager bash
apihw <hardware_id>
```

2. To generate api key for n smart devices

```
docker exec -it fleet_manager bash
apind <number of devices>
```

Setup plugin conveyor

1. Use the config editor, select the database plugin_config, select the collection plugin_conveyor, click on the document to edit it
2. Set activate_plugin to true

```
activate_plugin: true
```

3. Modify max_tote_per_trip if needed. This is the maximum number of totes that the sherpa can carry per trip
4. Restart FM
5. Conveyors need to be flashed along with fm_server_ip, cert_file and the right api key.

Setup plugin summon buttons

1. Use the config editor, select the database plugin_config, select the collection plugin_summon_button, click on the document to edit it
2. Set activate_plugin to true

```
activate_plugin: true
```

3. Restart FM
4. Flash summon buttons
5. What different colurs in summon button LED mean?

```
 "blinking red" - Not connected to FM
 "white" - Connected to FM but no unfinished trips
 "rotating yellow" - Trip booked on button press, waiting for the trip to
start
 "blinking green" - Waiting at station(Trip enroute)
 "rotating green" - Trip enroute
```

Flash summon buttons

1. Connect summon button to your laptop via USB to flash firmware
2. Download FlashTool_SB.tar from downloads section on the dashboard and run the same

```
tar -xvf FlashTool_SB.tar (This is for Linux, use similar commands to
extract files in other os)
cd FlashTool_SB
sudo bash ./install.sh
sudo bash ./flashtool_8mb.sh
```

3. Upon flashing, reconnect the summon button usb.
4. Press and hold the summon button until LED on the summon button turns blue, and connect to summon button via wifi. For instance you would see something like Summon_192049 in the available/known wifi networks. Upon successful connection to summon button wifi, you will see a summon button UI.
5. Press configure WiFi, choose the preferred network and add the wifi password for the same, save it. Wait unitl summon button led turns from yellow to blinking red .
6. Repeat step 4 and continue with the steps below
7. Now press configure device, add FM plugin url to HOST. PLUGIN_PORT by default would be 8002

```
ws://<FM_IP>:<PLUGIN_PORT>/plugin/ws/api/v1/summon_button
```

8. Set wifi type: WPA/WPA2

9. Set Mode to WiFi-Only
10. Set HEARTBEAT to disable
11. Set APIKEY and save.

```
X-API-Key:<api_key_generated_with_summon_button_id>
```

12. Press restart device in summon button UI.

Setup plugin IES

Will be added soon

Restart FM

1. If there were code changes/updates, 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
```

2. If there were only config changes, restart FM from the dashboard maintenance page

Debug FM

1. Check if there were any queue build ups. The output would show queue build ups if any.

```
docker exec -it fleet_manager bash
inspect ## This would list all historical queue build ups
rqi ## This would show if there is any queue build up at present
```

2. Check for occurences of rq errors (rqe) in fleet_manager.log, the output might lead to the issue

```
rq
```

3. If you are unable to login to FM, Check the docker logs - this should be run outside docker. There might be some errors in the init scripts.

```
docker logs fleet_manager
docker logs fleet_db
# Also check logs/fm.out inside fleet manager container
```

Access Postgres DB

1. Get a dump and copy it to the host machine, db_names can be ati_fleet, plugin_conveyor, plugin_summon_button, plugin_ies etc

```
docker exec -it fleet_db bash
pg_dump -U postgres <db_name> > /home/<db_name>.dump
exit
docker cp fleet_db:/home/<db_name>.dump .
```

2. Access db inside FM

```
docker exec -it fleet_manager bash
psql $FM_DATABASE_URI
```

3. Access db inside fm_plugins

```
docker exec -it fm_plugins bash
psql $PLUGIN_DATABSE_URI
```

Some docker commands

1. Useful docker commands (run outside container)

```
docker stats
docker system df
docker image prune
docker rmi <image_name>
docker stop <container_name>
docker rm <container_name>
```

Update FM with master FM credentials

1. Run the script mentioned below, you would need login credentials to sanjaya.atimotors.com to complete the update

```
 bash ./scripts/update_with_master_fm_cred.sh
```

2. There will be prompts to help you pull the images

```
Sanjaya Username: ### Enter master fm username ###
Sanjaya Password: ### Enter master fm password ###
FM version: ### Enter fm version like fm_dev, FM_v3.2 ###
```

3. Restart FM

Clean up disk space in FM server

1. Set backup config in FM config editor appropriately
2. Use the config editor, select the database fm_config, select the collection data_backup, click on the document to edit it
3. Edit keep_size_mb, FM will try to restrict the data inside static/data_backup folder to keep_size_mb only. The contents in the data backup folder will sorted and deleted based on their time of creation, older data will be deletedparking_id first. The default is set to 1000MB
4. Use can also set prune_ununsed_images to true or false based on whether you want to clean up old docker images. Set prune_images_used_until_h accordingly, all the images that were unused in the last prune_images_used_until_h hours will be deleted.

Create self signed certs for FM

This step needs to be done only if the certs were not setup already

1. 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]+'
```

2. 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 obtained in the previous step.

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

3. Restart FM (using docker-compose command)

Use trusted CA for FM

1. Create certificate bundle. The attached link can help you in creating cert bundle https://www.ssldragon.com/blog/what-is-a-ca-bundle/
2. Rename the certificate bundle as fm_rev_proxy_cert.pem
3. Rename key file as fm_rev_proxy_key.pem

4. Copy the renamed files(key, cert) to fm static/certs directory
5. Restart FM (using docker-compose command)
6. Set fm_cert_file in sherpa config.toml as below

```
fm_cert_file="/etc/ssl/certs/ca-certificates.crt"
```

RUN FM Simulator

FM simulator creates proxy for all the sherpa, Make sure real/physical sherpas are not connected to FM, Config changes need FM restart to take effect

1. Use the config editor, select the database fm_config, select the collection simulator, click on the document to edit it
2. Set simulate to true.

```
simulate: true
```

3. If you need to simulate visa/traffic gates as well, set visa_handling to true. **There is a caveat FM simulator would only request/release/simulate transit type visas, so full fledged visa simulation cannot be performed with fm simulator**

```
visa_handling: true
```

4. You can configure pre-defined routes, set book_trips to true. This can be used if there are so many trips to be booked. Route names don't not matter, but the stations list(route) must be valid. In the example below route1 is a schedule trip, route2 is a normal trip. In the route1 definition, 10 - trip frequency(in seconds), the following timestamps represent start_time and end_time respectively.

```
### OPTIONAL ###
book_trips: true,
routes : {
    "route1": [["Station A", "Station B"], ["10", "2023-05-31 15:00:00",
"2023-05-31 16:00:00"]],
    "route2": [["Station B", "Station A"], ["-1", "", ""]]
}
```

5. By default, sherpas will start at random stations. But if you want sherpas to start at specific stations use initialize_sherpas_at parameter. Follow the below example. Make sure the key, value match

sherpa name, station name exactly

```
### OPTIONAL ###
initialize_sherpas_at: {
    "sample_sherpa" : "Station A"
}
```

6. Restart FM

Where to get the logs?

We store logs corresponding to two containers (fleet_manager, fm_plugins)

1. Get fleet_manager logs

```
docker exec -it fleet_manager bash
cd logs
```

2. Get fm_plugins logs

```
docker exec -it fm_plugins bash
cd plugin_logs
```

3. Description of some important log files

   a. uvicorn.log - Any error related to fastapi app(endpoints) will be present in uvicorn.log

   b. fleet_manager.log - Any success/error in handlers/rq will be recorded in fleet_manager.log

   c. visa.log - All the visa assignments/rejects will be present in visa.log

   d. plugin_logs/plugin_<plugin_conveyor>/<plugin_<plugin_name>>.log - would have logs corresponding to the plugin

   e. plugin_logs/plugin_main.log - Any error in fm_plugins can be seen in plugin_main.log

Forgot password for frontend_user: admin ?

1. Use the config editor, select the database frontend_users, select the collection user_details
2. There would be multiple documents corresponding to different users, delete the document with name as admin
3. Restart FM
4. On restart you should be able to login in the below credentials

```
username: admin
password: 1234
```

Add new fleet

You would need ssh access to carry out this step

1. Create a folder in fleet_manager server static directory if not already present

```
## this needs to be done on
mkdir -p <fleet_manager_static_dir>/map
```

2. Copy the map files to the folder <fleet_manager_static_dir>/map/.
3. Use configure page in fleet_manager dashboard to add the fleet

Switch between multiple maps corresponding to the same fleet

These steps have to be carried out in the FM server

1. Create all_maps/ folder. You name version as you like.

```
For example, if there are two versions of map for the same fleet
mkdir -p <fleet_manager_static_dir>/all_maps/<version 1>
mkdir -p <fleet_manager_static_dir>/all_maps/<version 2>
```

2. Copy different sets of map files to the folders created above
3. Stop the fleet - using the option in the dashboard
4. Use dashboard, switch to corresponding fleet and press update_map button on the webpage header. A pop would help you to choose the required map version.
5. Start the fleet - using the option in the dashboard

Simulate summon_button/conveyor with postman

1. Create a new window for websocket connection in postman
2. Use the url given below based on plugin you are using

```
summon_button: ws://<fm_ip>:<plugin_port>/plugin/ws/api/v1/summon_button
conveyor: ws://<fm_ip>:<plugin_port>/plugin/ws/api/v1/conveyor
```

3. Set api-key in headers

```
X-API-Key: <api_key>
```

4. Send a tote_status message on behalf of conveyor. Only the attributes num_totes matters. All other are info given for debugging

```
{"sensor_1": 0,"sensor_2": 0,"sensor_3": 0,"sensor_4": 0,"sensor_5":
0,"sensor_6": 0,"num_totes": 1,"num_totes_to_transfer": 0,"compact_time":
0,"last_compact_time": 420774,"last_egress_time":
420798,"last_egress_reading": 0,"gate_state": 0,"gate_ready_time":
3421937,"last_tick_time": 28964247,"type": "tote_status"}
```

5. Simulate summon_button press, send button_pressed message

```
{"type": "button_pressed"}
```

Setup run on host service

This will be used to run commands outside docker's context, setting up this will require sudo access

1. Check the FM static dir, following files should exists

```
run_on_host.sh
run_on_host_updater.sh
install_run_on_host_service.sh
```

2. Give all perimissions to install_run_on_host_service script

```
sudo chmod ugo+rwx install_run_on_host_service.sh
```

3. Run install_run_on_host_service.sh

```
./install_run_on_host_service.sh
```

4. Ensure that the output shows that the run_on_host is successfully installed