**Complex Event Processing script setup**

**Setup Steps/Checks:**

1. Make sure directory ‘json’ is in same directory as ‘json\_to\_redis.sh’ and ‘json\_to\_redis.py’
2. Make sure ‘json\_to\_redis.sh’ and ‘json\_to\_redis.py’ are in the same directory
3. DB = 0 🡪 prod, DB = 1 🡪 dev on Redis
4. Need to do the **pip install geoip2**  on UI server or else maxmind python API will not work
   1. If pip is not already installed, make sure to install it
5. Need to do either : **sudo pip install redis**  or **sudo easy\_install redis**

To make sure redis-py client is installed, or else python redis connections will fail

1. **IF this is the first time pushing real-time scripts onto the UI server, and previously we have only been running sample scripts, THEN :**
   1. Follow these steps:
      1. redis-cli
      2. SELECT 0
      3. Del exec\_route\_info
2. Make sure that ‘GeoLite2-City.mmdb’ is in the same directory as ‘raw\_mtr\_parser.py’
3. Make sure redis server is up and running
   1. Try to do redis-cli – if redis tells you that it could not connect, you will have to restart the redis server
      1. nohup redis-server > ~/ redis.log & 2>&1
4. **IF YOU ARE RUNNING CEP ON LOCAL VM** 
   1. You will need to ‘cd ‘ into realtime\_dummy\_data
      1. Run python populate\_dummy.py
      2. This will populate Redis DB 0 raw\_mtr\_queue with 60,000 objects, and will also create historical values for them.
      3. You can ‘vim’ raw\_mtr\_queue and switch out the raw mtr lines with more current ones if you have access.
      4. Keep in mind, this will create two collections: tmp\_route\_historical, and raw\_mtr\_queue – it will render the script ‘json\_to\_redis.sh’ unnecessary
5. Schedule scripts as follows:
   1. **Git pull origin master**
   2. Need to add logs and necessary directories
      1. **Cd hourly**
         1. Mkdir logs
         2. Mkdir json
      2. **Cd streaming**
         1. Mkdir logs
      3. **Cd ETL/TWC/**
         1. Mkdir logs
      4. **Cd monit**
         1. Mkdir logs
      5. **Cd streaming**
         1. Mkdir logs
   3. **check if processes are already alive:**
      1. ps aux | grep python
      2. sudo kill -9 <process id : second column on line>
   4. **Start all streaming processes:**
      1. cd ./cep/streaming/
      2. nohup python set\_streaming\_on.py > logs/set\_streaming\_on.log & 2>&1
      3. nohup python raw\_mtr\_parser.py > logs/raw\_mtr\_parser.log & 2>&1
      4. nohup python populate\_exec\_collections.py > logs/populate\_exec\_collections.log & 2>&1
      5. nohup python populate\_live\_routes.py > logs/populate\_live\_routes.log & 2>&1
      6. nohup python populate\_sla\_route\_info.py > logs/populate\_sla\_route\_info.log & 2>&1
   5. **Start monit scripts:**
      1. Cd ./cep/monit
      2. Type in pwd to find the path to current position
      3. Vim monit.sh
         1. Add line at top: cd /path/to/cep/monit/
      4. Crontab –e
         1. Add \*/3 \* \* \* \* /path/to/monit.sh
   6. **Automate ETL scripts**
      1. Will automate to be running every other Sunday at 2 AM (every 2 weeks)
      2. Crontab –e
         1. Add 0 2 \* \* 0 /path/to/ETL/etl.sh
   7. **At the top of every hour we need to run the following scripts in THIS order:**
      1. python set\_streaming\_off.py
      2. python cleanup.py >> ./logs/cleanup.log
      3. python hourly\_averages.py
      4. sh json\_to\_redis.sh >> big\_data\_upload.log (this will call json\_to\_redis.py)
      5. python sla\_hourly.py
      6. python populate\_exec\_metric\_charts.py
      7. python populate\_sla\_ui.py
      8. python set\_streaming\_on.py
      9. **These are all placed in a shell script called ‘hourly\_executions.sh’. We need to add this to the crontab as follows:**
         1. First open up hourly\_executions.sh in an editor
            1. At the top, add the path to this file from home

Cd /home/path-to/cep/hourly/

* + - * 1. Save and close
      1. Cd into cep/hourly/
      2. Type in ‘pwd’ -- this will give you the present working directory
         1. Use this as the ‘file-path-to-hourly-exec’
      3. Crontab -e
         1. 00 \* \* \* \* ‘file-path-to-hourly-exec’/hourly\_executions.sh
  1. **Upon doing a git pull origin master(anytime when just updating code..)**
     1. You will need to make sure you go back to the following directories: hourly, monit, and ETL to:
        1. Open up the shell script that is part of a chron job, and check if there is a ‘cd ‘ command at the top, if there is **NOT**, then add to the top of the script ‘cd path/to/this/directory/…..for ex look at : step ‘fviii1a’