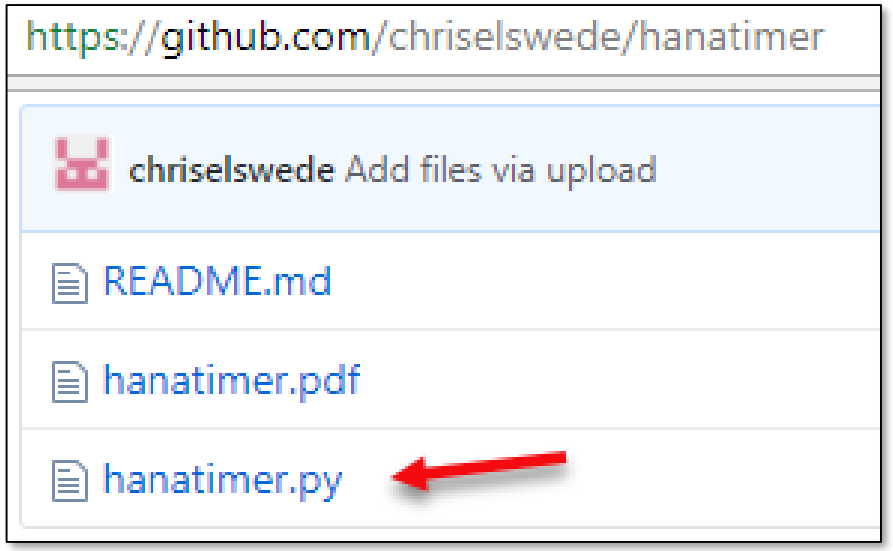
Monitoring

HANATimer – SAP Note 2634449

SAP Note 2634449 presents a tool that executes continuously a query and records the execution times

2634449 - How-To: Using SAP HANATimer

- It is a python script to be downloaded from https://github.com/chriselswede/hanatimer
- It is intended to be executed as <sid>adm on your SAP HANA Server
- It connects via host, port and DB user, provided in hdbuserstore



This script could be useful to understand if the runtime of a specific query

- is even over time or if there are time frames with significant time increases
- suffer from specific scenarios, e.g. increased delta storage or high resource consumption

Monitoring

HANATimer – using hdbuserstore

Host, port and DB user needs to be provided in the hdbuserstore:

```
oqladm@ls80010:/> hdbuserstore SET HANATIMERKEY ls80010:30015 HANATIMER Passwd1234
oqladm@ls80010:/> hdbuserstore LIST HANATIMERKEY
KEY HANATIMERKEY
  ENV : ls80010:30015
  USER: HANATIMER
```

Then the hanatimer can connect using the info stored in hdbuserstore:

```
oqladm@ls80010:/tmp/HANATimer> python hanatimer.py -k HANATIMERKEY -sql "select * from dummy"
Start Time,             Overall Time [micro seconds],          Server Time [micro seconds]
2018-04-19 18:00:24,    866,                                   202
2018-04-19 18:00:25,    728,                                   204
2018-04-19 18:00:26,    750,                                   203
2018-04-19 18:00:27,    902,                                   209
```

HANATimer – needs privileges

The DB user that hanatimer uses to connect with needs enough privileges to execute the query given by the -sql flag

HANATimer – SQL Query

HANATimer have to be provided a SQL query

Flag	Details	Explanation	Default
-sql	test query	this sql statement will be executed and timed, the statement has to be quoted, i.e. surrounded by two "	<none>

Example:

```
oqladm@ls80010:/tmp/HANATimer> python hanatimer.py -k HANATIMERKEY -sql "select * from dummy"
Start Time,             Overall Time [micro seconds],          Server Time [micro seconds]
2018-04-19 18:00:24,    866,                                   202
2018-04-19 18:00:25,    728,                                   204
2018-04-19 18:00:26,    750,                                   203
2018-04-19 18:00:27,    902,                                   209
```

The output lists

- **Overall time;** for SAP HANA server processing, network communication and client processing
- **Server time;** for SAP HANA server processing

Monitoring

HANATimer – Wait Time and Test Period

HANATimer has to be told how long to wait between each executions and for how long it will continuously execute the sql statement and record the execution times

Flag	Unit	Details	Explanation	Default
-ws	seconds	wait	how many seconds hanatimer waits after it is done with one execution to start next execution	1
-tp	hours	test period	how many hours hanatimer will continously execute and time the sql statement	1

Example:
Here hanatimer waits 5 seconds after an execution finished until it starts the execution again and it will do this for 3 hours

```
oqladm@ls80010:/tmp/HANATimer> python hanatimer.py -k HANATIMERKEY -sql "select * from dummy" -ws 5 -tp 3
Start Time,            Overall Time [micro seconds],          Server Time [micro seconds]
2018-04-19 18:15:02,   886,                                   212
2018-04-19 18:15:07,   761,                                   201
2018-04-19 18:15:12,   1179,                                   189
2018-04-19 18:15:17,   821,                                   200
```

HANATimer – Output

HANATimer can be told in what directory to store the log files and if it should write to standard out or not

Flag	Unit	Details	Explanation	Default
-od		output directory	full path of the folder where all output files will end up (if the folder does not exist it will be created)	/tmp/hanatimer_output
-so	true/false	standard out switch	switch to write to standard out	true

Example:
Here hanatimer creates the new directory for the log and does not write to standard out:

```
oqladm@ls80010:/tmp/HANATimer> rm -r /tmp/hanatimerout/
oqladm@ls80010:/tmp/HANATimer> python hanatimer.py -k HANATIMERKEY -sql "select * from dummy" -od /tmp/hanatimerout/ -so false


^CTraceback (most recent call last):
  File "hanatimer.py", line 131, in <module>
    main()
  File "hanatimer.py", line 127, in main
    time.sleep(float(wait_seconds))
KeyboardInterrupt
oqladm@ls80010:/tmp/HANATimer>
oqladm@ls80010:/tmp/HANATimer> more /tmp/hanatimerout/hanatimerlog_2018-04-19_18-23-40.csv
Start Time,            Overall Time [micro seconds],         Server Time [micro seconds]
2018-04-19 18:23:40,   779,                                  204
2018-04-19 18:23:41,   713,                                  188
2018-04-19 18:23:42,   800,                                  192
```