## **Single-beat**

Single-beat is a nice little application that ensures only one instance of your process runs across your servers.

Such as celerybeat (or some kind of daily mail sender, orphan file cleaner etc...) needs to be running only on one server, but if that server gets down, well, you go and start it at another server etc.

As we all hate manually doing things, single-beat automates this process.

Docs: <https://github.com/ybrs/single-beat>

## **Settings Needs to understand:**

**SINGLE\_BEAT\_IDENTIFIER:**

* the default is we use your process name as the identifier
  + This would be the process name, which will be used by all servers.
* all processes checks a key named, SINGLE\_BEAT\_<value>.

**SINGLE\_BEAT\_LOCK\_TIME**

* (default 5 seconds)
* This is the TTL time for the key, which has been set by one of your server.

**SINGLE\_BEAT\_INITIAL\_LOCK\_TIME:**

* (default 2 \* SINGLE\_BEAT\_LOCK\_TIME seconds)
* when starting your process, single-beat set a key with 10 second expiration (INITIAL\_LOCK\_TIME) in redis server.
* other single-beat processes checks if that key exists - if it exists they won't spawn children.

**SINGLE\_BEAT\_HEARTBEAT\_INTERVAL**

* (default 1 second)
* Single-beat continue to update that key every 1 second (HEARTBEAT\_INTERVAL) setting it with a ttl of 5 seconds (LOCK\_TIME)

**SINGLE\_BEAT\_HOST\_IDENTIFIER**

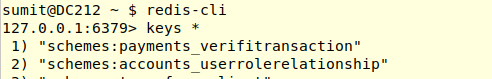
* (default socket.gethostname)
* It should vary for each server

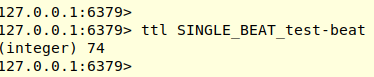
*This should work, but you might want to give more relaxed intervals, like:*

Ex: SINGLE\_BEAT\_LOCK\_TIME=300 SINGLE\_BEAT\_HEARTBEAT\_INTERVAL=60 single-beat celery beat

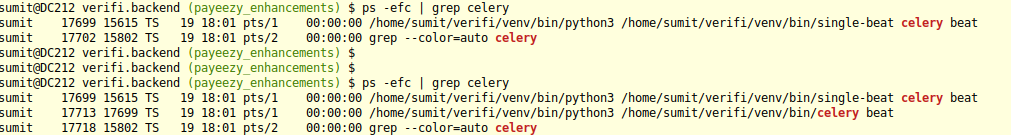
**Testing Work flow (POC):**

**Pre-requisites:**

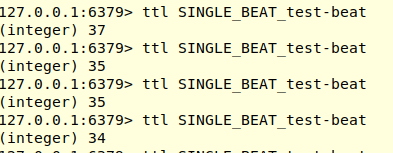
1. Redis should be up and running
2. Singlebeat should be installed.
3. Hitting command
   1. “SINGLE\_BEAT\_HOST\_IDENTIFIER='host-1' SINGLE\_BEAT\_IDENTIFIER='test-beat' SINGLE\_BEAT\_LOCK\_TIME=40 SINGLE\_BEAT\_HEARTBEAT\_INTERVAL=15 single-beat celery beat”
   2. This will start single-beat process on machine and wait for 60 seconds to check heart-beat.
   3. After 15 seconds it will check redis for key “SINGLE\_BEAT\_test-beat” and becuase it does not exists, it will create a key, set value and set ttl of (2\*40) 80 seconds on it.







4. Once key is set, it will regularily update the key and updates its ttl. (this time ttl time would be SINGLE\_BEAT\_LOCK\_TIME=40 ).



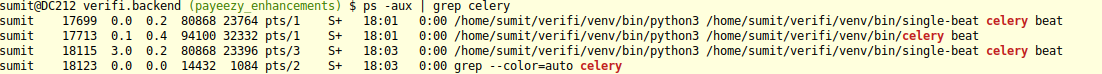
5. Now starts another server with command:

(Same BEAT\_IDENTIFIER but different SINGLE\_BEAT\_HOST\_IDENTIFIER)

SINGLE\_BEAT\_HOST\_IDENTIFIER='host-2' SINGLE\_BEAT\_IDENTIFIER='test-beat' SINGLE\_BEAT\_LOCK\_TIME=40 SINGLE\_BEAT\_HEARTBEAT\_INTERVAL=15 single-beat celery beat

6. This will only start single-beat process, not celery beat, because each time its

cheking redis for key (it exists)



7. Case (if server 1 got crashed and failed)

Key will expire when its ttl isexhausted.

After key expires and heart-beat interval completed for server2, server 2 will check for key and as its does not exists, it will spawn new children..

