

Mun Sentinel Setup Guide

Sentinel is a python script that runs a series of tests on the masternode to make sure it is functioning properly and reports its status to the network. Absence of sentinel may cause the status changed from ENABLED to WATCHDOG_EXPIRED followed by NEW_START_REQUIRED. While under the WATCHDOG_EXPIRED your node will still get paid it does not constitute normal behavior and you will need to install a sentinel. If your node is ENABLED and runs fine you do not need to install sentinel script until further notice when the governance and superblocks are enabled and proper voting rights will be essential.

This guide covers the installation of the sentinel on a VPS running Ubuntu 16.04 where your MN is setup.

Make sure the mun.conf is fully configured as described in the masternode setup guide and contains rpc user name, password, and a line "server=1" that allows for the node to listen to rpc commands otherwise you will get "Connection refused" error.

Check the python version and make sure 2.7.x is installed:

```
python --version
```

The output should be 2.7.x, for example:

```
Python 2.7.12
```

Clone sentinel script in a new directory under your home directory and switch to it:

```
git clone https://github.com/muncrypto/sentinel.git
```

Switth to sentinel directory:

```
cd sentinel
```

Update system packages:

```
sudo apt-get update
```

Install python virtualenv, which is a way of preserving and restoring python environment in a separate folder including all binaries and packages:

```
sudo apt-get -y install python-virtualenv
```

Create a python virtual environment folder under the sentinel folder:

```
virtualenv ./venv
```

Restore the python environment locally:

```
./venv/bin/pip install -r requirements.txt
```

Edit the location of your mun.conf file **only if it is not at the default location**, such as when you use the -datadir=... option, then change it for both the main script as well as for the test script:

In file sentinel.conf uncomment and modify line:

```
mun_conf=/home/YOURUSERNAME/YOURDATAPATH/mun.conf
```

In file test/test_sentinel.conf insert the same line to specify the path to mun.conf.

Test the sentinel configuration by executing the test script:

```
./venv/bin/py.test ./test
```

The output should be:

```
===== test session starts =====  
platform linux2 -- Python 2.7.12, pytest-3.0.1, py-1.4.31, pluggy-0.3.1  
rootdir: /home/roma/muncoin1/bin/sentinel, inifile:  
collected 14 items
```

```
test/integration/test_jsonrpc.py .  
test/unit/test_misc.py .  
test/unit/test_models.py ..  
test/unit/test_mun_config.py .  
test/unit/test_mund_data_shims.py ..  
test/unit/test_muny_things.py .....
```

```
test/unit/test_submit_command.py .
```

```
===== 14 passed in 0.16 seconds =====
```

Add the sentinel to your periodic task scheduler so that it runs every 5 minutes a command to get into your sentinel directory and run a python script without reporting any errors:

```
*/5 * * * * cd /home/YOURUSERNAME/sentinel && ./venv/bin/python bin/sentinel.py  
>/dev/null 2>&1
```

To view debug output set the sentinel environment variable to anything non-zero then run the script:
SENTINEL_DEBUG=1 ./venv/bin/python bin/sentinel.py