lvcgcn

Dec 07, 2020

Contents:

This is the documentation for the TOROS responder to the Gamma- ray burst Coordinates Network (GCN) for Gravitational-Wave (GW) alerts from the LIGO-Virgo Collaboration (LVC).

The responder lvcgcnd is a daemon managed by the systemd service found in most Linux systems.

CHAPTER 1

Installation and Usage

1.1 Installation

To install the lvcgcnd systemd service type in a terminal:

```
$ make
$ sudo make install
```

The first command will install the Python package `torosgcn` and a script `lvcgcn`. It will also create a `lvcgcnd.service` file to create the systemctl daemon `lvcgcnd`.

The second command needs root permissions. It will copy the `lvcgcnd.service` file to `/etc/systemd/system` and it will also copy a sample `lvcgcn-conf.yaml` file to `/etc/toros` (see section *Configuration*).

It is highly recommended (but not necessary) that you use a virtual environment when installing.

1.2 Usage

Usage of the `lvcgcnd` daemon follow all of the rules for `systemd` services.

For a full documentation of the systemd service, please refer to the official documentation.

1.2.1 Most common commands

To start running the daemon:

```
$ systemctl start lvcgcnd
```

If you change the configuration, you can restart the daemon to use the new config:

```
$ systemctl restart lvcgcnd
```

To stop it:

```
$ systemctl stop lvcgcnd
```

If you want to check if `lvcgcnd` whether is active and running or not:

```
$ systemctl status lvcgcnd
```

All the previous commands may require `root` privilege to execute.

CHAPTER 2

Workflow

Here we describe the workflow of `lvcgcnd` after an alert is received.

`lvcgcnd` is a systemd wrapper for a Python script (`lvcgcn`) created at install time. When `lvcgcnd` is started it will setup a logger and release control to pygcn specifying a callback function (`process_gcn`) to be called when an alert is received.

2.1 Processing Steps

The sequential list of operations after an alert is received is encapsulated in the method `torosgcn.listen.process_gcn` and can be summarized as follows:

- Parse VOEvent and retrieve information for specific keywords.
- If the alert is a mock test and `DEBUG_TEST` is on, pass it along, otherwise return.
- Backup VOEvent to an XML file in the filesystem, if required in the configuration.
- Send the alert email to people in `Alert Recipients`.
- Retrieve skymap (if any) from GraceDB website.
- Save to file a copy of skymap if required in the configuration.
- Generate targets for each observatory using the skymap and GLADE catalog.
- Upload GCN Notice information and targets, if any, to broker website.

If any of these steps fails with error, the custom loguru logging system will send an email to the admins specified in the configuration file.

2.2 Directory Structure

lvcgcn will make use of the following files and directories:

- `/etc/systemd/system/lvcgcnd.service`: The systemd service for lvcgcn daemon.
- `/etc/lvcgcn/lvcgcn-conf.yaml`: The configuration file used by lvcgcn.
- `/var/lvcgcn/log`: Default directory to store logs. It can be changed in the configuration file.
- `/var/lvcgcn/skymaps`: Default directory to store FITS of sky maps.
- `/var/lvcgcn/VOEvents`: Default directory to store VOEvent XML files.
- `/etc/lvcgcn/GLADE_2.3.csv`: Default place where lvcgcn will look for GLADE catalog CSV file.

CHAPTER 3

Configuration

Lvcgcn makes use of a configuration file installed in /etc/toros/lvcgcn-conf.yaml

It uses the YAML mark-up language.

Note: During installation, a sample `lvcgcn-conf.yaml` is installed to help with configuration.

3.1 Keyword Description

Following is a description of the keyworks that you should configure

`LIGO Run`: A string for the LIGO Run name (O1, O2, O3, etc).

`Catalog Path`: The full path to the galaxy catalog file to retrieve sources. lvcgcnd is set to work with GLADE 2.3. If you want to modify the catalog used, see section *Galaxy Catalogs*.

`Catalog Filters`: Parameters to filter galaxy selection criteria. This depends on the particular selection criteria and are used in `torosgcn.scheduler.generate_targets`.

`Observatories`: lvcgcnd will generate separate lists of targets for each observatory based on what each can observe. This keyword should contain a list and each item should look like the following:

```
name: OBS01
location: {
  lon: -64.5467, # degrees
  lat: -31.5983, # degres
  height: 1350   # meters
}
```

`DEBUG_TEST`: If `true` it will respond to mock alerts (the M-series). It will only send emails to people specified under `Admin Emails`, but it will upload the targets to the broker website and will backup the files if specified. Settign it to `false`, will ignore them.

`Email Configuration`: A dictionary with the sender email configuration. It should look like the following:

```
SMTP Domain: smtp.gmail.com:587,
Sender Address: example@gmail.com,
Login Required: true,
Username: yourUserName,  # null if not needed
Password: $ecretPassw0rd,  # null if not needed
```

`Admin Emails`: A yaml list with the list of the administrators of lvcgcnd. Admins will be alerted of error when `DEBUG_TEST` is set to `true`.

`Alert Recipients`: A yaml list with the recipient emails that should be notified when an alert is received.

`Broker Upload`: A dictionary containing URLs and credentials to upload targets for different observatories. An example is given below.

```
site url: https://toros.utrgv.edu/,
login url: https://toros.utrgv.edu/account/login/,
uploadjson url: https://toros.utrgv.edu/broker/uploadjson/,
logout url: https://toros.utrgv.edu/account/logout/,
username: admin,
password: Adm1nPa$$word
```

`Logging`: The file path (`File`) and `Log Level` (`DEBUG`, `INFO`, `WARNING`, `ERROR`, `CRITICAL`) to be set for logging.

`Backup`: Whether to backup VOEvents and skymap files.

CHAPTER 4

JSON Sample File

The GCN Notice information and targets for observatories is uploaded to the broker site using a JSON file through a private API. Below is a sample JSON file.

The `assignments` key is optional ("Retraction" notices do not specify assignments).

```
{
  "alert": {
    "ligo_run": "O3",
    "graceid": "S190501a",
    "SEtype": "S",
    "datetime": "2019-06-01T03:53:23"
  },
  "gcnnotice": {
    "gcntype": "Initial",
    "datetime": "2019-06-01T04:21:31"
  },
  "assignments": {
    "OBS01": {
      "NGC3527": 4.579577594439791e-07,
      "SDSSJ111036.33+291631.5": 4.571471198600905e-07,
      "SDSSJ110931.04+290153.1": 4.570529667464678e-07,
      "SDSSJ105717.80+262039.7": 4.565977915949241e-07,
      "UGC06147": 4.5358279508508366e-07
    },
    "OBS02": {
      "1790410": 4.754568836051921e-07,
      "1751874": 4.729037759899382e-07,
      "SDSSJ105332.98+254611.7": 4.727430685333189e-07,
      "SDSSJ105324.63+254607.4": 4.7274177982139006e-07,
      "1831688": 4.674778642517513e-07,
    },
    "OBS03": {
      "SDSSJ105530.54+260442.7": 4.6561621329489456e-07,
      "SDSSJ110509.50+283702.6": 4.629186351956917e-07,
      "SDSSJ110619.55+282301.5": 4.608629463520819e-07,
```

(continues on next page)

(continued from previous page)

```
      "NGC3527": 4.579577594439791e-07,
      "UGC06147": 4.5358279508508366e-07
    }
  }
}
```

CHAPTER 5

Galaxy Catalogs

lvcgcn is designed to work with the GLADE catalog (revision 2.3).

To speed up the reading process `torosgcn.scheduler.get_targets` will need a comma separated value (csv) reduced version of GLADE.

The columns for the reduced version should at least contain `Name`, `RA`, `Dec`, and `Dist`.

lvcgcnd will open the catalog using the `Catalog Path` entry in the `lvcgcn-conf.yaml` configuration file.