Relay Commander Documentation

Release 0.0.6

LaunchDarkly Solutions Engineering Team

Mar 05, 2019

Contents

A CLI for managing LaunchDarkly Relay Instances.

CHAPTER 1

Quickstart

1.1 Quickstart

1.1.1 Installation

You can install the latest version of relaycommander with:

```
pip install relaycommander
```

CHAPTER 2

Runbook

2.1 RunBook

RelayCommander is a CLI tool that is intended to allow you to manually make a change to feature flags should your applications lose connectivity with the LaunchDarkly service. You must have the LD Relay setup with Redis in order for the CLI tool to work. It works by manually updating the status of a flag directly within Redis, while at the same time recording each update that has taken place. Then, once connection has been re-established with LaunchDarkly, you will then run a command that will update your configuration back to our service via the API. This iteration allows you to change the state of a feature flag to either ON or OFF. Due to the current way that SDK's work with redis, this can only be used to update the status of a backend feature flag.

2.1.1 Setup

- You can install relay commander with `pip install relaycommander`; this will enable the rc command globally.
- LD Relay proxy with Redis is setup
- Backend SDK clients are connected to the relay box

2.1.2 Instructions

Pre-requisites

- **Create a `.env` file similar to the sample file and be sure to update the following:**
  - **REDIS_HOSTS**
    - Update the `.env` file to include the host name(s) and port of the redis instances.

    If there are multiple redis instances running, provide as a CSV list of host names.
  - **LD_API_KEY**

- LaunchDarkly API token to be used when writing the updates back to LaunchDarkly.

  Note that the API token requires administrative proviliges in order to work.

While there is a disconnect with LaunchDarkly

Update redis by running:

```
rc update -p project -e environment -f feature_flag -s state
```

- Project = the key of the project to be updated
- Environment = the key of the environment to be updated
- Feature = the key of the feature to be updated
- State = the state of the feature flag you would like to change it to. Currently allows you to set it to on or off

Each time this command is run, we will create a new direcoty called playback with a file containing the corresponding that needs to be run using the API

- No changes required: Changes will take effect once the relay cache detects the update and will broadcast the update to the SDK clients
- (Optional) Restart the relay proxy to server to make the updates immediate

Warning: NO CHANGES CAN BE MADE WHILE LAUNCHDARKLY IS A DISCONNECTED. IT IS RECOMMENDED THAT YOU EITHER HAVE AN INTERNAL PROCESS SO THAT NO ONE MAKES UPDATES DURING THIS TIME OR YOU DISABLE ALL LOGINS VIA SSO

Once LD is reconnected

Run the following command:

```
rc replay
```

Running this command will iterate through all of the files that were created during `rc update` and make the corresponding udpates in LaunchDakly via the API to synch it with the offline changes that were made

Verify that the current state in LaunchDarkly matches that last state that was set using RelayCommander

CHAPTER 3

CLI Reference

3.1 CLI

3.1.1 CLI Interface

Relay Commander CLI

CHAPTER 4

API Reference

If you are looking for information on a specific function, class or method, this part of the documentation is for you.

4.1 API

4.1.1 Utilities

Generator Module.

Generates templates using jinja.

class relay_commander.generators.**ConfigGenerator**
Abstract configuration generator using Jinja

generate_relay_config(*environments*)
Generate ld-relay.conf file.

validator module

Helper functions to validate CLI input

relay_commander.validator.**validateState**(*state*)
Validate State Argument

Checks that either 'on' or 'off' was entered as an argument to the CLI and make it lower case.

Parameters **state** – state to validate

relay_commander.replayBuilder.**checkLocal**()
Check for directories and create if not already there

relay_commander.replayBuilder.**createFile**(*project*, *environment*, *feature*, *state*)
Create file with RC command that will be called against LaunchDarkly API.

Parameters

- **Project** – LaunchDarkly Project

- **environment** – LaunchDarkly Environment
- **feature** – LaunchDarkly Feature
- **state** – State to update feature flag

relay_commander.replayBuilder.**executeReplay**(*logger=None*)
Execute commands

1. Itereate through files created in ./replay/toDO
2. Execute the command inside each file
3. Move the file to the archive.

4.1.2 Wrappers

ld module

Wrapper for the LaunchDarkly API

Reference API - https://pypi.org/project/launchdarkly-api/

class relay_commander.ld.**LaunchDarklyApi**(*apiKey*, *projectKey=None*, *environmentKey=None*, *logger=None*)
Wrapper for the LaunchDarkly API

formatHostname(*key*)
Returns formatted hostname for an environment.

Parameters **key** – environment key

getEnvironments(*projectKey*)
Returns List of Environments for a Project.

Includes name, key, and mobile key, and formatted hostname.

Parameters **projectKey** – Key for project

Returns Collection of Environments

updateFlag(*state*, *featureKey*)
Update the flag status for the specified feature flag

Parameters

- **state** – New feature flag state
- **featureKey** – Feature flag key

Returns boolean status of the feature flag attribute "on"

relay_commander.redis

A Redis Wrapper

class relay_commander.redis.**RedisConention**(*host*, *port*)
Redis Connetion

Parameters

- **host** – hostname for redis
- **port** – port for redis

class relay_commander.redis.**RedisWrapper**(*host*, *port*, *logger*, *projectKey*, *environmentKey*)
Relay Specific Redis Wrapper.

Parameters

- **projectKey** – LaunchDarkly project key
- **environmentKey** – LaunchDarkly environment key.
- **conn** – (optional) redis connection string

_formatKeyName()
Return formatted redis key name.

static connectionStringParser(*uri*)
Parse Connection string to extract host and port.

Parameters **uri** – full URI for redis connection in the form of

host:port

Returns list of RedisConnection objects

getFlagRecord(*featureKey*)
Get feature flag record from redis.

Parameters **featureKey** – key for feature flag

updateFlagRecord(*state*, *featureKey*)
Update redis record with new state.

Parameters

- **state** – state for feature flag
- **featureKey** – key for feature flag

CHAPTER 5

Developer Documentation

5.1 Developer Documentation

5.1.1 Installing Development Environment

This project uses pipenv for dependency management.

1. Make sure you have pipenv installed locally.
2. Clone this repo and run `pipenv install -e .`
3. Run some cli commands with `pipenv run rc`

5.1.2 Running Tests

Tests can be found in the `tests` directory.

You can run tests with `make tests`.

If you want to run a specific test file you can do so with:

```
python -m unittest tests/relay_commander/test$MODULE.py
```

Code Coverage

This project attempts to have 100% code coverage. when you run `make test` code coverage is automatically ran. You can view the code coverage report locally by opening up the index.html file in the `htmlcov` directory that gets created when you run `make test`.

5.1.3 Documentation

This project uses sphinx for documentation. You can generate the latest docs locally by running `make docs`. You can then view them by opening up the `index.html` file in the `docs/build/html` directory.

5.1.4 Linting and Style

This project follows the PEP 8 style guidelines. You can install `pylint` in order to ensure that all of your code is compliant with this standard.

CHAPTER 6

Changelog

6.1 Changelog

Here you can see the full list of changs between each relaycommander release.

6.1.1 Version 0.0.11

Released on March 5, 2019

Warning: This release fixes a critical bug where commands were not automatically being replayed via the CLI. All users are urged to upgrade to this version as soon as possible if it is being used in production.

- Refactor CLI to accept `on` or `off` instead of `true` or `false`. This better aligns with the experience that a user would have if they were to flip a kill switch in the LaunchDarkly UI.
- Fix a critical bug where commands were not being replayed properly via the playback command.
- Add support for python 3.5, 3.6, and 3.7 - due to an upstream issue with swagger and python 3.7 we were not able to use the LD API wrapper in versions greater than 3.6. This is no longer the case.
- Add integration tests to test out the general flow of the run book on all supported python versions during CI.
- Add new command to generate a relay proxy configuration for a given project.

6.1.2 Version 0.0.10

Released on January 28, 2019

- Add feature that allows users to override the default redis port as

a part of the configuration.

6.1.3 Version 0.0.9

Released on January 25, 2019

- Add feature to allow users to define multiple redis destinations. The CLI

will then attempt to update all hosts and report back the status via the console.

Warning: All of the versions below have been unpublished from pypi to reduce confusion. If you need a specific version from below you can download the appropriate git tag and build the package from there.

6.1.4 Version 0.0.8

Released on January 17, 2019

- Configure Logging
- Fix bug that failed to update redis due to missing key

6.1.5 Version 0.0.7

Released on January 16, 2019

- Fix bug in CLI command that does not allow you to update redis.

6.1.6 Version 0.0.6

Released on January 16, 2019

- Added sphinx documentation

6.1.7 Version 0.0.5

Released on Dec 19, 2018

- First alpha release including base functionality, code coverage, and unit tests.

Python Module Index

r

Index

Symbols

C

E

F

G

L

R

U

V