
SatNOGS Config

Release 0.5.4+0.gb5e8711.dirty

SatNOGS

Dec 29, 2020

CONTENTS

1	Table of Contents	3
1.1	User guide	3
1.1.1	Requirements	3
1.1.2	Installation	3
1.1.3	Usage	3
1.1.3.1	Basic	3
1.1.3.2	Advanced	3
1.1.3.3	Show	4
1.1.3.4	Upgrade	4
1.1.3.5	Update	4
1.1.3.6	Reset	4
1.1.3.7	Apply	4
1.2	Developer guide	4
1.2.1	Installation	4
1.2.2	Configuration	4
1.2.3	Extending	4
1.2.4	Modifying the menu	5
1.2.5	Code Quality Assurance	5
1.2.6	Automation	5
1.3	Modules reference	5
1.3.1	satnogsconfig	5
1.3.2	satnogsconfig.config	5
1.3.3	satnogsconfig.helpers	6
1.3.4	satnogsconfig.menu	7
2	Indices and tables	9
	Python Module Index	11
	Index	13

SatNOGS Config is a utility for generating the configuration for a Debian-based SatNOGS client system. It is a menu driven configuration tool built on-top of `pythondialog`. The main purpose of this utility is to create the YAML configuration file for SatNOGS Client Ansible which provisions the system. In addition to that, it interacts with `satnogs-setup`, which is a wrapper to bootstrap Ansible, upgrades the system packages using APT and probes the hardware to show the user a more limited set of configuration options.

TABLE OF CONTENTS

1.1 User guide

1.1.1 Requirements

- Python 3.4+

1.1.2 Installation

To install SatNOGS Config run:

```
$ pip install satnogs-config
```

This will install a console script called `satnogs-config`.

1.1.3 Usage

To execute the script, run it on the command line:

```
$ satnogs-config
```

The main menu will be shown

1.1.3.1 Basic

The `Basic` sub-menu contains the minimum required configuration options for provisioning the SatNOGS client system.

1.1.3.2 Advanced

The `Advanced` sub-menu contains a larger set of configuration options. It is intended for advanced users and contains options categorized by component.

1.1.3.3 Show

The `Show` option displays the current configuration file.

1.1.3.4 Upgrade

The `Upgrade` option calls APT utilities to upgrade the distribution packages.

1.1.3.5 Update

The `Update` option works in conjunction with `satnogs-setup` to re-bootstrap Ansible and update the configuration tools.

1.1.3.6 Reset

The `Reset` option resets the configuration file generated by the utility, effectively removing all the configuration options set by the user.

1.1.3.7 Apply

The `Apply` button runs SatNOGS Client Ansible which provisions Debian to become a SatNOGS client system.

1.2 Developer guide

1.2.1 Installation

To install SatNOGS Config for development run in the project root directory:

```
$ pip install -e .
```

1.2.2 Configuration

This project uses `python-dotenv`. Configuration of `satnogsconfig/settings.py` can be overridden by setting the respective environment variables or an `.env` file placed on the project root directory.

1.2.3 Extending

SatNOGS Config functionality can be extended by implementing additional helpers. The helpers are used to enhance menu functionality beyond the core function of generating a YAML file for Ansible.

1.2.4 Modifying the menu

The menu itself is also expressed in YAML format. The menu structure is defined in `menu.yml` file and shipped along with the package.

1.2.5 Code Quality Assurance

The following code quality assurance tools are used in this project:

- flake8
- isort
- yapf
- pylint

1.2.6 Automation

tox is used to automate development tasks. To execute the default list of tasks run:

```
$ tox
```

1.3 Modules reference

1.3.1 satnogsconfig

1.3.2 satnogsconfig.config

Configuration module

class `satnogsconfig.config.Config(filename)`
Manage configuration file

Parameters `filename` (*str*) – File path of configuration

`_load_config()`

Load and parse YAML configuration

Returns Configuration dictionary

Return type dict or NoneType

`clear_config()`

Clear configuration file

`dump_config(to_file=False)`

Dump configuration in YAML format

Parameters `to_file` (*bool, optional*) – Dump to file

Returns YAML configuration

Return type str

`get_variable(variable)`

Get variable value from configuration

Parameters **variable** (*str*) – Variable to get the value

Returns Value of variable

Return type *str* or *bool* or *NoneType*

set_variable (*variable*, *value*)

Set variable value in configuration

Parameters

- **variable** (*str*) – Variable to set the value
- **value** (*str* or *bool* or *NoneType*) – Value of variable

1.3.3 satnogsconfig.helpers

Helpers module

class `satnogsconfig.helpers.SatnogsSetup`

Interact with satnogs-setup

property `gr_satnogs_version`

Get installed gr-satnogs version

Returns Version of gr-satnogs

Return type *str*

property `is_applied`

Check whether configuration has been applied

Returns Whether configuration has been applied

Return type *bool*

static `restart (boot=False)`

Restart satnogs-setup script

Parameters **boot** (*bool*, *optional*) – Whether to bootstrap or not

property `satnogs_client_ansible_version`

Get installed SatNOGS Client Ansible version

Returns Version of SatNOGS Client Ansible

Return type *str*

property `satnogs_client_version`

Get installed SatNOGS Client version

Returns Version of SatNOGS Client

Return type *str*

property `tags`

Get satnogs-setup tags

Returns Set of tags

Return type *set*

upgrade_system ()

Upgrade system packages

class satnogsconfig.helpers.GrSatnogs

Get and set gr-satnogs configuration

property gr_satnogs_version

Get gr-satnogs version

probe_soapysdr ()

Probe SoapySDR hardware

class satnogsconfig.helpers.Ansible (*ansible_dir*)

Call Ansible playbooks

pull (*playbooks, url, branch=None, tags=None, extra_args=None*)

Pull and run Ansible playbook

Parameters

- **playbooks** (*list*) – List of playbooks
- **url** (*str*) – Git URL to pull playbooks
- **branch** (*str, optional*) – Git branch to pull playbooks
- **tags** (*list, optional*) – List of tags
- **extra_args** (*list, optional*) – List of extra arguments to pass to Ansible

Returns Whether Ansible execution succeeded

Return type bool

run (*playbooks, tags=None, extra_args=None*)

Run Ansible playbook

Parameters

- **playbooks** (*list*) – List of playbooks
- **tags** (*list, optional*) – List of tags
- **extra_args** (*list, optional*) – List of extra arguments to pass to Ansible

Returns Whether Ansible execution succeeded

Return type bool

1.3.4 satnogsconfig.menu

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

S

`satnogsconfig.config`, [5](#)
`satnogsconfig.helpers`, [6](#)

Symbols

`_load_config()` (*satnogsconfig.config.Config method*), 5

A

`Ansible` (*class in satnogsconfig.helpers*), 7

C

`clear_config()` (*satnogsconfig.config.Config method*), 5

`Config` (*class in satnogsconfig.config*), 5

D

`dump_config()` (*satnogsconfig.config.Config method*), 5

G

`get_variable()` (*satnogsconfig.config.Config method*), 5

`gr_satnogs_version()` (*satnogsconfig.helpers.GrSatnogs property*), 7

`gr_satnogs_version()` (*satnogsconfig.helpers.SatnogsSetup property*), 6

`GrSatnogs` (*class in satnogsconfig.helpers*), 6

I

`is_applied()` (*satnogsconfig.helpers.SatnogsSetup property*), 6

P

`probe_soapysdr()` (*satnogsconfig.helpers.GrSatnogs method*), 7

`pull()` (*satnogsconfig.helpers.Ansible method*), 7

R

`restart()` (*satnogsconfig.helpers.SatnogsSetup static method*), 6

`run()` (*satnogsconfig.helpers.Ansible method*), 7

S

`satnogs_client_ansible_version()` (*satnogsconfig.helpers.SatnogsSetup property*), 6

`satnogs_client_version()` (*satnogsconfig.helpers.SatnogsSetup property*), 6

`satnogsconfig.config` (*module*), 5

`satnogsconfig.helpers` (*module*), 6

`SatnogsSetup` (*class in satnogsconfig.helpers*), 6

`set_variable()` (*satnogsconfig.config.Config method*), 6

T

`tags()` (*satnogsconfig.helpers.SatnogsSetup property*), 6

U

`upgrade_system()` (*satnogsconfig.helpers.SatnogsSetup method*), 6