

---

# **SatNOGS Config**

*Release 0.9+1.gb51cd13.dirty*

**SatNOGS**

**Mar 27, 2020**



# CONTENTS

<b>1</b>	<b>Table of Contents</b>	<b>3</b>
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	6
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



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

### 1.3.4 satnogsconfig.menu

## INDICES AND TABLES

- genindex
- modindex
- search



## PYTHON MODULE INDEX

### S

`satnogsconfig.config`, 5



## Symbols

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

## 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

## S

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

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