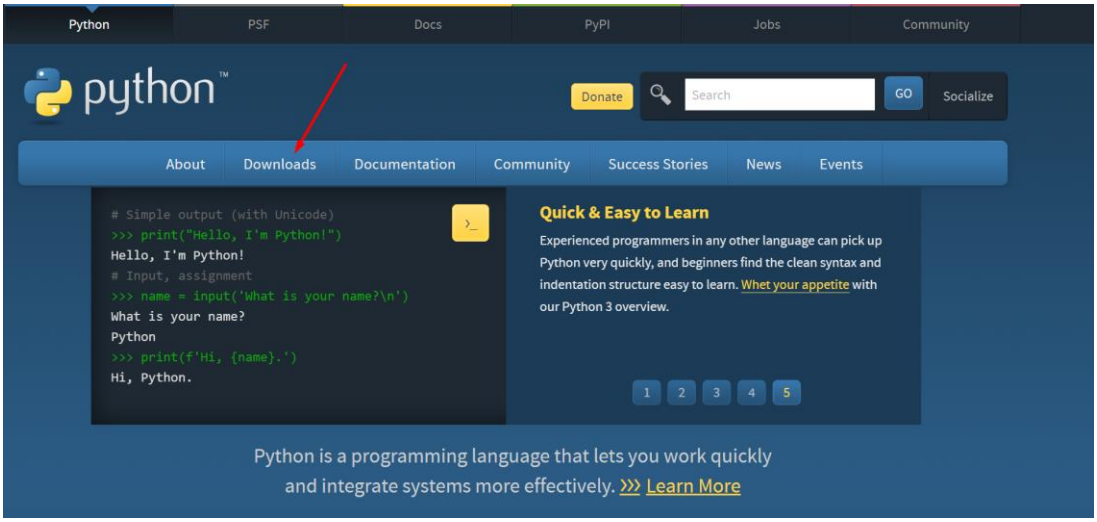
CONTENT

SETUP

1. Download and setup Python (recommended version = 3.10.9). Dont forget to add Python to PATH. You can download Python from official website (https://www.python.org)

   Here are some screenshots to help you install the app correctly

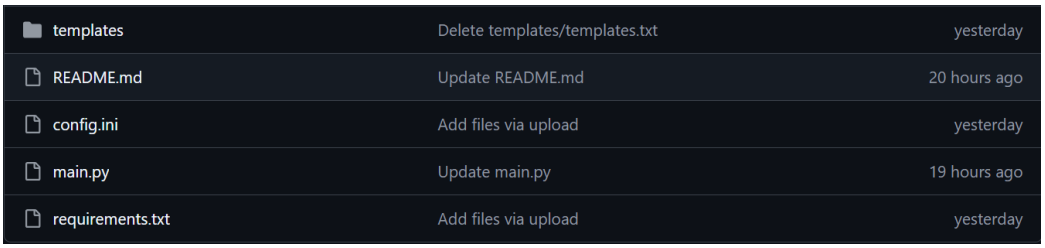


- Python 3.10.9 - Dec. 6, 2022

  Note that Python 3.10.9 *cannot* be used on Windows 7 or earlier.

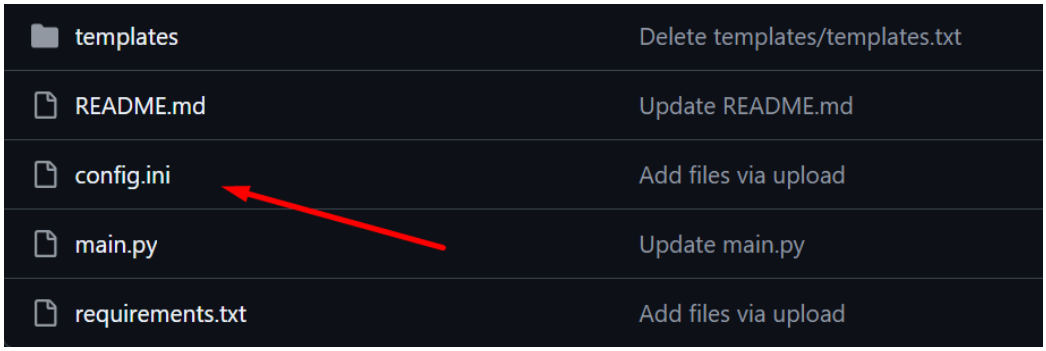
  - Download Windows installer (64-bit)
  - Download Windows installer (32-bit)
  - Download Windows help file
  - Download Windows embeddable package (64-bit)
  - Download Windows embeddable package (32-bit)

2. Download this project. The project is available on GitHub with all the necessary files such as:



3. Install requirements by command - "pip install -r requirements.txt"(run command inside the project folder directory)

4. Change personal data in config.ini file.




```
[General]
refresh_delay = 180
host = 999.999.99.99
login = LOGIN
password = PASSWORD
local_host = 127.0.0.1
local_port = 5000
```

FUNCTIONAL DOCUMENTATION:

Additional Libraries:

Connect to the switch threw library **netmiko** (Network automation to screen-scraping devices is primarily concerned with gathering output from show commands and with making configuration changes.)

Netmiko aims to accomplish both of these operations and to do it across a very broad set of platforms. It seeks to do this while abstracting away low-level state control (i.e. eliminate low-level regex pattern matching to the extent practical)

Link - https://ktbyers.github.io/netmiko/docs/netmiko/index.html

---

Connect to the Web interface threw library **Flask** (This is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications)

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

Link - https://flask.palletsprojects.com/en/stable/

Netmiko	Network automation to screen-scraping devices is primarily concerned with gathering output from show commands and with making configuration changes.	https://ktbyers.github.io/netmiko/docs/netmiko/index.html
Flask	This is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications	https://flask.palletsprojects.com/en/stable/

Common python Libraries

Sys

This module in Python provides access to variables and functions that interact with the Python interpreter. It allows you to handle system-specific parameters and functions and can be very helpful in managing how Python code interacts with the operating system.

Time

This module in Python provides a set of functions to handle time-related tasks, such as getting the current time, pausing execution, or converting time formats. It is useful when you need to measure the performance of code, delay operations, or work with timestamps.

Threading

This module in Python is used to create and manage threads, which allow you to run multiple operations concurrently in a program. This is particularly useful for tasks that may block execution, like I/O operations or when you want to improve the responsiveness of an application.

Re

This module in Python provides support for working with regular expressions, which are a powerful tool for searching and manipulating strings

based on specific patterns. Regular expressions allow you to match, search, split, and replace text in a flexible and efficient way.

Configparser

This module in Python is used to handle configuration files. These files often have the .ini extension and are widely used to store settings and options for applications. The configparser module allows you to read, write, and modify configuration files easily.

Logging

This module in Python is a standard way to log messages from your application. It is a powerful and flexible system that can be used to record a wide variety of events, from simple debug statements to critical error messages. The logs can be directed to different outputs, such as the console, files, or even remote servers.

OS

This module in Python provides a way of using operating system-dependent functionality, such as interacting with the file system, managing environment variables, and performing tasks like running shell commands. It serves as a bridge between Python and the underlying operating system, allowing you to perform various system-level operations.

CONSOLE COMMANDS:

Commands that we execute in server terminal to get port information

- ***display interface brief***

The command **display interface brief** on a Huawei switch is used to display a summary of the status and basic information of all the interfaces on the device. This command provides an easy way to check the status of the interfaces at a glance and is helpful for network troubleshooting and monitoring.

```
Interface           PHY    Protocol    IP Address      Description
------------------------------------------------------------------
GE1/0/1             up     up          192.168.1.1     ---
GE1/0/2             down   down        --              ---
10GE1/0/1           up     up          10.0.0.1        ---
```

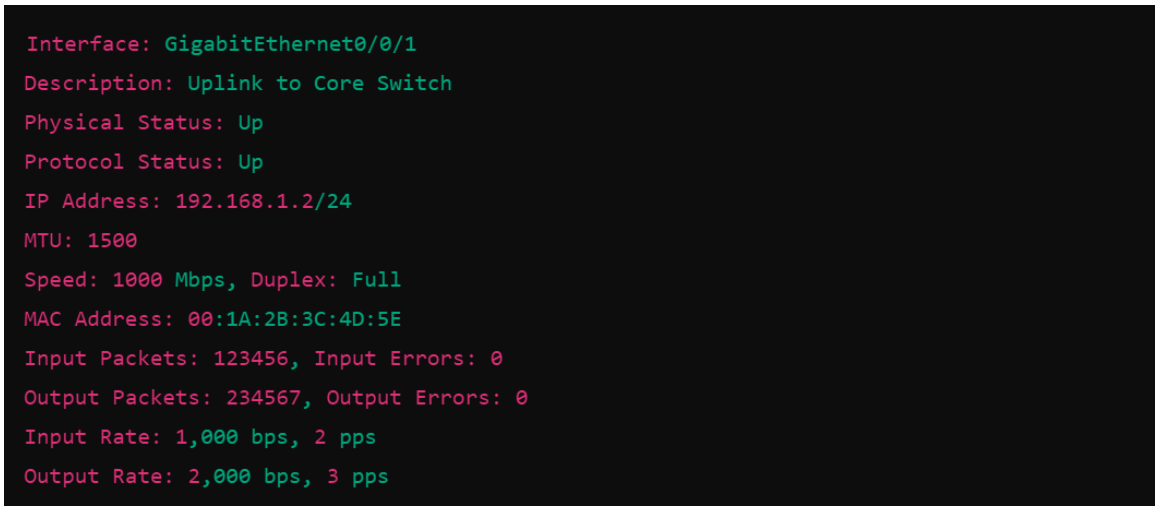
- ***display transceiver***

The command **display transceiver** on a Huawei switch is used to show detailed information about the transceivers (optical modules) installed in the switch. This includes specifics about the transceivers' status, properties, and performance data. Transceivers are devices such as SFP, QSFP, or XFP modules that are used to connect network devices via optical or copper cables.

```
Port      : XGigabitEthernet0/0/1
Type      : 10GBase-LR
Vendor    : HUAWEI
Wavelength: 1310nm
Temperature: 40°C
Voltage   : 3.3V
Current   : 50mA
Tx Power  : -1.5dBm
Rx Power  : -2.0dBm
Serial Number: ABC123456789
```

- ***display interface***

The command **display interface** on a Huawei switch is used to show detailed information about one or all network interfaces on the device. This command provides in-depth status and performance data, which can help with diagnosing and monitoring the network interfaces



PROGRAM FUNCTIONS:

- ***get_port_info***

The get_port_info function is designed to extract and return information related to a specific network port from a given block of text (data).

- ***analyze_ports***

The analyze_ports function is designed to assess the health and performance of network ports based on various criteria. It takes in a list of port data and analyzes key attributes such as physical status, protocol status, utilization rates, and error counts to categorize the ports into three groups: good, bad, and inactive.

- ***Get_massive***

The Get_massive function is designed to process a raw output string, typically containing tabular data (like the output from a command-line interface), and convert it into a list of dictionaries. Each dictionary represents a row of data with key-value pairs corresponding to headers and their associated values.

- ***create_connection***

The create_connection function is designed to establish a connection to a Huawei network device using the SSH protocol (typically facilitated by libraries such as Netmiko or similar). It configures connection parameters and manages the connection process.

- ***execute_command***

The execute_command function is designed to send a command to a network device over an established connection and return the output of that command. This function typically works in conjunction with a library like Netmiko, which facilitates communication with network devices.

- ***clean_up_info***

The clean_up_info function is designed to process a string by removing excessive whitespace, making the text cleaner and easier to read or process further. It uses regular expressions to achieve this.

- ***parse_transceiver_info***

The parse_transceiver_info function is designed to extract and structure information from the raw output of a network device's transceiver status. This function processes text data by using regular expressions to identify and clean relevant sections, organizing the information into a list of dictionaries.

- ***fetch_switch_data***

The fetch_switch_data function is a key part of a network management system, designed to establish a connection to a network switch, execute commands to retrieve data, and parse and analyze that data in a loop.

- ***index***

The index function is part of a web application, likely built using Flask, a popular web framework for Python. This function is responsible for handling requests to the main index page of the application.

- ***read_config***

The read_config function is designed to read configuration settings from a file, specifically using the configparser module in Python.

CONFIGURATION SETTINGS:

refresh_delay

This setting typically refers to a parameter that determines the time interval (in seconds) at which a system, application, or component refreshes or updates its data. This is often relevant in contexts where data changes frequently or where the application needs to periodically retrieve new information from a source (such as a database, API, or hardware device).

---

host = 999.999.99.99

This setting typically refers to the address of a server or device that an application needs to connect to. This parameter is essential in networked applications where communication with other systems, databases, or services is required. The host setting specifies the endpoint for these connections and can take various forms depending on the context.

---

login = LOGIN

In configuration settings for SSH, the login parameter typically refers to the username that will be used to authenticate when connecting to a remote server via SSH.

---

password = PASSWORD

In configuration settings for SSH, the password parameter refers to the password associated with the username specified in the SSH connection settings.

This password is used during the authentication process to establish a secure connection to a remote server.

---

local_host = 127.0.0.1

This particular IP address is commonly associated with the localhost, which refers to the local computer on which the application is running.

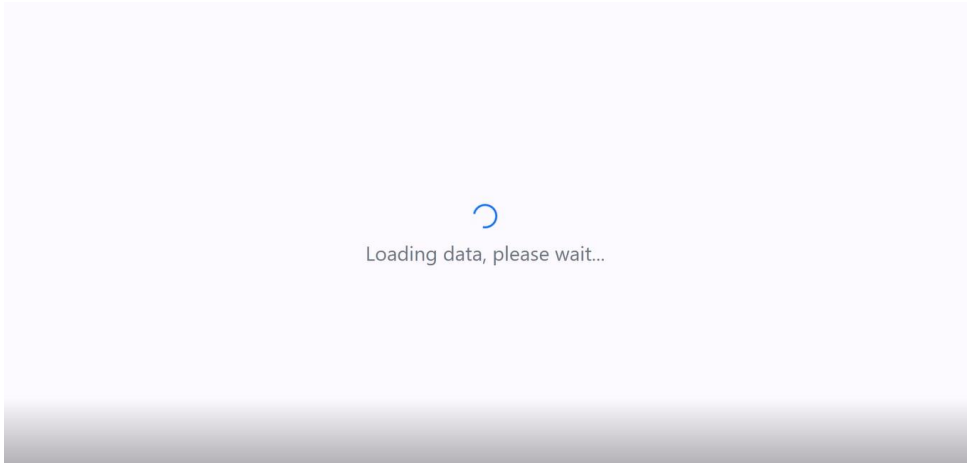
---

local_port = 5000

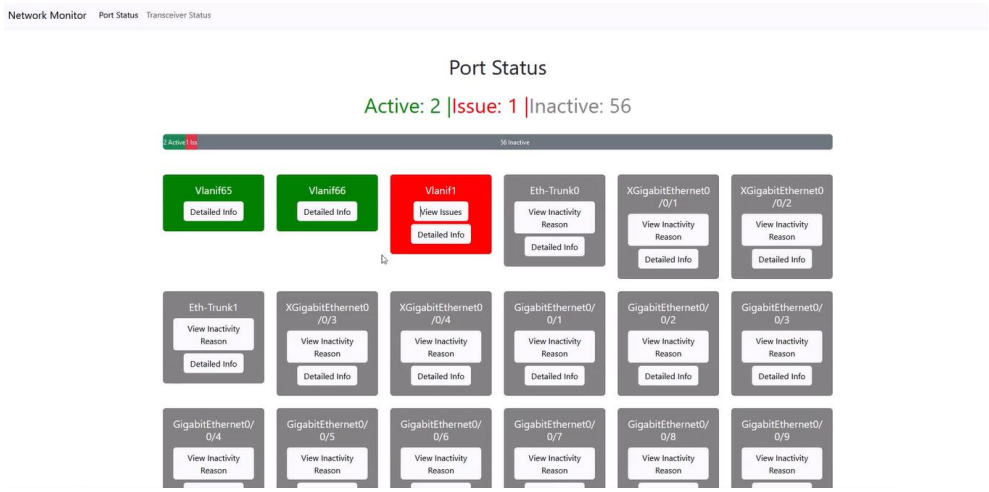
This value typically represents the port number that a service, application, or server will listen on for incoming connections.

USER MANUAL

1. INSTALL our program by “SETUP INSTRUCTION”
2. Optional: Connect to the VPN
3. Open cmd and go to project directory
4. Run “python main.py”
5. Open webpage provided by program on local-ip configured by you

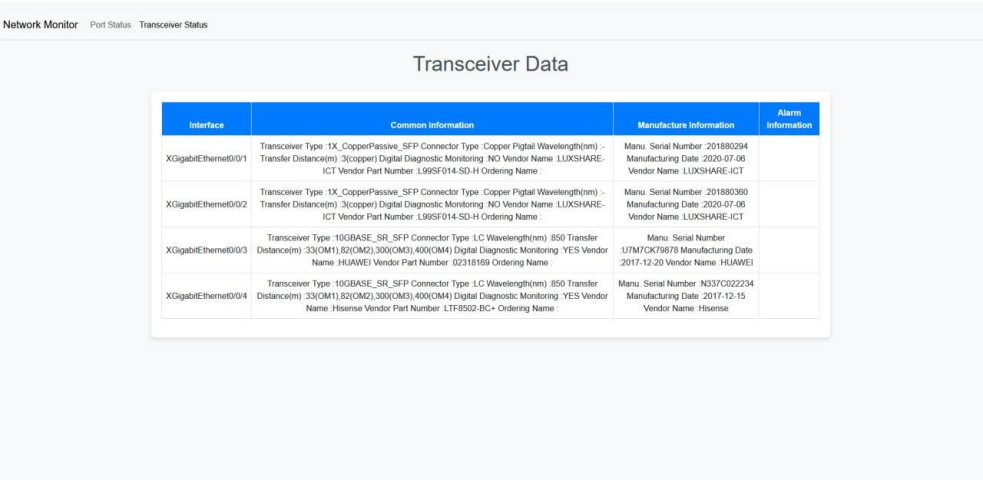


6. Wait until program display port statistic (If waiting continue for more than 2 minutes, check cmd for error messages)



7. View interface information (which contains detailed port information, issue port information, inactivity ports information, count of active ports, issue ports, inactive ports)

8. View Transceiver information by clicking “Transceiver Status” title on navbar



9. Optional: Check program logger for port information change data