Web Scraping Project

Table of Contents

Introduction

This project scrapes data from a specified website and stores it in a structured format. It uses a distributed system of master and worker nodes, configured using Tailscale, to handle the scraping tasks.

Prerequisites

- Python 3.x
- Tailscale account
- Web scraping libraries (BeautifulSoup, Scrapy, Selenium, etc.)

- Database (MySQL, MongoDB, etc.)

Installation

Web Scraping Tools

1. Install Python and pip:

   sudo apt-get install python3

   sudo apt-get install python3-pip

2. Install necessary libraries:

   pip3 install requests beautifulsoup4 scrapy selenium

Tailscale

1. Install Tailscale on your master and worker nodes:

   curl -fsSL https://tailscale.com/install.sh | sh

Web Scraping Process

Step-by-Step Flow

1. Identify Target Website:

   - Choose the website and ensure compliance with its terms of service.

2. Understand Website Structure:

- Use browser developer tools to inspect HTML elements.

3. Select a Scraping Tool or Library:

- Options include BeautifulSoup, Scrapy, Selenium, Puppeteer, etc.

4. Install Necessary Dependencies:

- Ensure all required packages are installed.

5. Fetch HTML Content:

- Use the scraping tool to send HTTP requests and retrieve HTML content.

6. Parse HTML Content:

- Extract specific elements using the chosen library.

7. Handle Dynamic Content:

- Use tools like Selenium for dynamically loaded content.

8. Data Cleaning and Transformation:

- Clean and transform data as needed.

9. Store Data:

- Save data to a file, database, or cloud storage.

10. Handle Pagination:

- Implement a mechanism to navigate and scrape paginated content.

Master and Worker Nodes Setup

Step-by-Step Setup

1. Log in to Tailscale:

   - Log in using your Tailscale account credentials.

2. Initialize the Master Node:

   - On the master node machine, initialize Tailscale:

     tailscale up

3. Authorize the Master Node:

   - Follow the authorization process as instructed.

4. Retrieve Master Node IP Address:

   - Get the Tailscale IP address of the master node:

     tailscale ip

5. Install and Initialize Worker Nodes:

   - On each worker node, install and initialize Tailscale:

     tailscale up --authkey <master_node_auth_key>

6. Authorize Worker Nodes:

   - Follow the authorization process for each worker node.

Running the Scraper

1. Ensure all nodes are authorized and connected via Tailscale.

2. Run the scraping script on the master node, which will distribute tasks to worker nodes.

Data Storage

- Data can be stored in a local file, database (MySQL, MongoDB), or cloud storage.

Handling Pagination

- Implement logic to navigate through paginated content and scrape data from each page.

Contributing

Contributions are welcome! Please create a pull request or open an issue to discuss any changes.

License