

Open-Source-Tool: Auto SS Config

Ich freue mich, bekannt geben zu können, dass ich ein Tool namens **Auto SS Config** open-source gestellt habe. Dieses Tool generiert und lädt automatisch Shadowsocks- oder Clash-Abonnement-URLs aus Shadowsocks-URLs hoch, was die Verwaltung und Aktualisierung Ihrer Proxy-Server-Konfigurationen erleichtert.

Dieses Tool war ein echter Game-Changer für mich, besonders wenn mein Shadowsocks-Server blockiert wird. Ich verwende den Outline Manager, um einen neuen Server zu erstellen, eine frische Adresse zu erhalten und diese URL direkt über die Mac-App zu importieren, um die GFW-Beschränkungen zu umgehen. Das Ausführen von `python upload_configs.py` aus diesem Projekt aktualisiert meine Abonnement-URLs und stellt sicher, dass alle meine digitalen Geräte funktionierende Netzwerkverbindungen behalten.

Funktionen

- **Konvertiert Shadowsocks-URLs in Clash-Konfigurationen:** Einfacher Wechsel zwischen verschiedenen Proxy-Konfigurationen.
- **Unterstützt mehrere Shadowsocks-Server:** Verwalten Sie mehrere Server mit Leichtigkeit.
- **Lädt Konfigurationen automatisch in Google Cloud Storage hoch:** Halten Sie Ihre Konfigurationen sicher und zugänglich.
- **Macht Konfigurationen öffentlich zugänglich:** Teilen Sie Ihre Konfigurationen mit anderen.
- **Verwendet Cache-Control für sofortige Updates:** Stellen Sie sicher, dass Ihre Konfigurationen immer auf dem neuesten Stand sind.

Dateien

- `app_config_tmp.yaml`: Anwendungskonfiguration (Bucket-Name, SS-URLs).
- `clash_config_tmp.yaml`: Temporäre Clash-Konfigurationsdatei.
- `upload_configs.py`: Skript zur Erstellung der Clash-Konfiguration und zum Hochladen der Konfigurationen in Google Cloud Storage.
- `requirements.txt`: Python-Abhängigkeiten.

Einrichtung

1. **Abhängigkeiten installieren:** `bash pip install -r requirements.txt`

2. **Google Cloud-Anmeldeinformationen einrichten:**

- Installiere das Google Cloud SDK.
- Führe den Befehl `gcloud auth application-default login` aus.
- Oder setze die Umgebungsvariable `GOOGLE_APPLICATION_CREDENTIALS`.

3. **Kopieren Sie** `app_config_tmp.yaml` **nach** `app_config.yaml` **und konfigurieren Sie es:**

```
yaml bucket_name: Ihr-Bucket-Name ss_urls: - ss://methode:password@server:port
```

Verwendung

1. **Fügen Sie Ihre Shadowsocks-URLs zur** `ss_urls`-**Liste in** `app_config.yaml` **hinzu:** `yaml`

```
ss_urls: - ss://method:password@server:port
```

2. **Konfigurationen hochladen:** `bash python upload_configs.py`

Das Skript gibt die öffentlichen URLs für beide Konfigurationen aus.

Entwicklung

- **Python 3.6+**
- Verwendet `ruamel.yaml` für die YAML-Verarbeitung.
- Verwendet `google-cloud-storage` für GCS-Operationen.

Lizenz

MIT (Massachusetts Institute of Technology)

Schauen Sie sich gerne das Repository an, um weitere Details zu erfahren und um Beiträge zu leisten!