Gestion des IP réservées sur DigitalOcean

Il est courant que les adresses IP des serveurs soient facilement bloquées par le Grand Firewall (GFW) de Chine. Cela est particulièrement vrai pour les serveurs cloud. Pour atténuer ce problème, une stratégie consiste à utiliser les IP réservées de DigitalOcean et à les réassigner à votre droplet lorsque l'IP actuelle est bloquée. Ce post présente un script Python pour automatiser ce processus. Le script est également open-source et disponible sur GitHub.

Le script vous permet de :

- Vérifier si une IP réservée est assignée à un droplet spécifique.
- Réassigner une nouvelle IP réservée à un droplet si l'IP actuelle est bloquée.
- Vérifier si le port 80 est ouvert sur l'IP réservée (une manière simple de vérifier si l'IP fonctionne).

Voici le script Python :

```
import socket
import os
import argparse
import json
import requests
import time

# Fonction pour obtenir les en-têtes de l'API DigitalOcean
def get_digitalocean_headers():
    api_key = os.environ.get("DO_API_KEY")
    if not api_key:
        print("Erreur : DO_API_KEY introuvable dans les variables d'environnement.")
        return None
    return {
        "Authorization": f"Bearer {api_key}",
        "Content-Type": "application/json"
    }

# Fonction pour récupérer toutes les IP réservées de DigitalOcean
def fetch_reserved_ips():
```

```
    headers = get_digitalocean_headers()
    if not headers:
        return None
    try:
        url = "https://api.digitalocean.com/v2/reserved_ips"
        resp = requests.get(url, headers=headers)
        resp.raise_for_status()
        reserved_ips_data = resp.json().get("reserved_ips", [])
        with open('response.json', 'w') as f:
            json.dump(reserved_ips_data, f, indent=4) # Sauvegarde la réponse dans un fichier pour le d
        return reserved_ips_data
    except requests.exceptions.RequestException as e:
        print(f"Erreur lors de la récupération des adresses IP réservées : {e}")
        return None

# Fonction pour désassigner une IP réservée d'un droplet
def unassign_ip_from_droplet(ip_address, droplet_id, droplet_name):
    headers = get_digitalocean_headers()
    if not headers:
        return False

    try:
        url = f"https://api.digitalocean.com/v2/reserved_ips/{ip_address}"
        resp = requests.delete(url, headers=headers)
        resp.raise_for_status()
        print(f"IP {ip_address} désassignée avec succès du droplet {droplet_name}")
        return True
    except requests.exceptions.RequestException as e:
        print(f"Erreur lors de la désassignation de l'IP {ip_address} du droplet {droplet_name} : {e}")
        return False

# Fonction pour assigner une IP réservée à un droplet
def assign_ip_to_droplet(ip_address, droplet_id, droplet_name):
    headers = get_digitalocean_headers()
    if not headers:
        return False
```

```
    try:
        url = f"https://api.digitalocean.com/v2/reserved_ips/{ip_address}/actions"
        req = {
            "type": "assign",
            "droplet_id": droplet_id
        }
        resp = requests.post(url, headers=headers, json=req)
        resp.raise_for_status()
        print(f"IP {ip_address} assignée avec succès au droplet {droplet_name}")
        return True
    except requests.exceptions.RequestException as e:
        print(f"Erreur lors de l'assignation de l'IP {ip_address} au droplet {droplet_name} : {e}")
        return False

# Fonction pour traiter les IP réservées, vérifier leur assignation et les réassigner si nécessaire
def process_reserved_ips(reserved_ips, droplet_name, only_check=False):
    if not reserved_ips:
        print("Aucune IP réservée trouvée dans votre compte.")
        return None

    for reserved_ip in reserved_ips:
        ip_address = reserved_ip.get("ip")
        if not ip_address:
            print("Aucune adresse IP trouvée pour une IP réservée.")
            continue

        droplet = reserved_ip.get("droplet", None)
        if droplet_name:
            if droplet and droplet.get("name") == droplet_name:
                print(f"L'IP réservée {ip_address} est assignée au droplet : {droplet_name}")
                if only_check:
                    if check_port_80(ip_address):
                        print(f"Le port 80 est ouvert sur {ip_address} pour le droplet {droplet_name}")
                    else:
                        print(f"Le port 80 est fermé sur {ip_address} pour le droplet {droplet_name}")
```

```
                    return ip_address
                droplet_id = droplet.get("id")
                if droplet_id:
                    if unassign_ip_from_droplet(ip_address, droplet_id, droplet_name):
                        # Tentative d'assigner une nouvelle IP après la désassignation

                        new_ip = create_new_reserved_ip(droplet_id)
                        if new_ip:
                            print("Attente de 10 secondes avant d'assigner la nouvelle IP...")
                            time.sleep(10)
                            if assign_ip_to_droplet(new_ip, droplet_id, droplet_name):
                                print(f"Nouvelle IP {new_ip} assignée avec succès au droplet {droplet_na
                            else:
                                print(f"Échec de l'assignation de la nouvelle IP {new_ip} au droplet {d
                        else:
                            print("Aucune IP disponible à assigner")

                else:
                    print(f"Impossible de désassigner l'IP {ip_address} car l'ID du droplet n'a pas été
                return None
            elif droplet:
                print(f"L'IP réservée {ip_address} n'est pas assignée au droplet : {droplet_name}")
            else:
                print(f"Aucun droplet n'est assigné à l'IP réservée : {ip_address}")
        else:
            return ip_address
    return None

# Fonction pour créer une nouvelle IP réservée
def create_new_reserved_ip(droplet_id):
    headers = get_digitalocean_headers()
    if not headers:
        print("Échec de la récupération des en-têtes DigitalOcean.")
        return False
    try:
        url = "https://api.digitalocean.com/v2/reserved_ips"
```

```
            req = {
                "region": "sgp1", # Vous pouvez changer la région si nécessaire
            }
            print(f"Tentative de création d'une nouvelle IP réservée pour le droplet ID : {droplet_id}")
            resp = requests.post(url, headers=headers, json=req)
            resp.raise_for_status()
            new_ip = resp.json().get("reserved_ip", {}).get("ip")
            print(f"Nouvelle IP réservée créée avec succès : {new_ip}")
            return new_ip
        except requests.exceptions.RequestException as e:
            print(f"Erreur lors de la création d'une nouvelle IP réservée : {e}")
            return False

# Fonction pour vérifier si le port 80 est ouvert sur une adresse IP
def check_port_80(ip_address):
    try:
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
            s.settimeout(5)
            s.connect((ip_address, 80))
            return True
    except Exception:
        return False

# Fonction principale pour obtenir l'IP réservée
def get_reserved_ip(droplet_name=None, only_check=False):
    reserved_ips = fetch_reserved_ips()
    if reserved_ips is None:
        return None
    return process_reserved_ips(reserved_ips, droplet_name, only_check)

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description="Obtenir une adresse IP réservée de DigitalOcean.")
    parser.add_argument("--droplet-name", required=True, help="Nom du droplet pour vérifier si l'IP rés
    parser.add_argument("--only-check", action="store_true", help="Vérifie uniquement si l'IP est assig
    args = parser.parse_args()
```

```
reserved_ip = get_reserved_ip(args.droplet_name, args.only_check)
if reserved_ip:
    print(f"L'adresse IP réservée est : {reserved_ip}")
```

Explication :

1. **Importation des bibliothèques :** Importe les bibliothèques nécessaires pour les opérations réseau, les variables d'environnement, l'analyse des arguments, la gestion du JSON, les requêtes HTTP et les délais temporels.
2. `get_digitalocean_headers()` **:** Récupère la clé API de DigitalOcean à partir des variables d'environnement et construit les en-têtes nécessaires pour les requêtes API.
3. `fetch_reserved_ips()` **:** Récupère toutes les IP réservées associées à votre compte DigitalOcean via l'API. Elle sauvegarde également la réponse brute dans `response.json` pour le débogage.
4. `unassign_ip_from_droplet()` **:** Désassigne une IP réservée donnée d'un droplet spécifié.
5. `assign_ip_to_droplet()` **:** Assigne une IP réservée donnée à un droplet spécifié.
6. `process_reserved_ips()` **:** C'est la logique principale :
   - Elle parcourt toutes les IP réservées.
   - Si un `droplet_name` est fourni, elle vérifie si l'IP est assignée à ce droplet.
   - Si `only_check` est vrai, elle vérifie si le port 80 est ouvert et retourne l'IP.
   - Si `only_check` est faux, elle désassigne l'IP actuelle, en crée une nouvelle et assigne la nouvelle IP au droplet.
7. `create_new_reserved_ip()` **:** Crée une nouvelle IP réservée dans la région `sgp1` (vous pouvez la changer).
8. `check_port_80()` **:** Vérifie si le port 80 est ouvert sur une adresse IP donnée. C'est une manière simple de vérifier si l'IP est accessible.
9. `get_reserved_ip()` **:** Orchestre le processus de récupération et de traitement des IP réservées.
10. `if __name__ == '__main__':` **:** Analyse les arguments de ligne de commande (`--droplet-name` et `--only-check`) et appelle `get_reserved_ip` pour exécuter le script.

Comment utiliser :

1. **Configurer la clé API DigitalOcean :** Définissez la variable d'environnement `DO_API_KEY` avec votre clé API DigitalOcean.
2. **Exécuter le script :**

- Pour vérifier si une IP est assignée à un droplet et si le port 80 est ouvert : `bash python votre_nom_de_script.py --droplet-name votre_nom_de_droplet --only-check`
- Pour réassigner une nouvelle IP à un droplet : `bash python votre_nom_de_script.py --droplet-name votre_nom_de_droplet`

Ce script fournit un cadre de base pour la gestion des IP réservées. Vous pouvez l'étendre en fonction de vos besoins spécifiques.