Configurer un VPN client sécurisé pour une utilisation dans un conteneur Docker

Table des matières

Introduction

Conditions préalables

- Exigences
- Composants utilisés
- Informations de licence

Configuration

- Fichier Docker

Introduction

Ce document décrit comment utiliser le VPN Cisco Secure Client à l'intérieur d'un conteneur Docker.

Conditions préalables

Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Le package Cisco Secure Client peut être téléchargé sur le bureau local et utilisé dans un conteneur Docker. (Pour télécharger le package client, reportez-vous à la page Web Cisco Secure Client.)
- Cisco Secure Client est compatible avec Docker à partir de la version 5.1.10.
- Le déploiement de Docker nécessite l'utilisation des packages Cisco Secure Client DEB ou RPM CLI (les packages sont optimisés pour une utilisation CLI uniquement, ce qui est le cas pour Docker).

Composants utilisés

Les informations contenues dans ce document sont basées sur la version 5.1.10 RPM de Cisco Secure Client ou sur le package DEB CLI.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations de licence

Reportez-vous au Guide de commande Cisco Secure Client pour obtenir des informations sur les licences.

Configuration

Fichier Docker

1. Installation du package dont dépend le client sécurisé Cisco.
    - Pour RHEL (Red Hat Enterprise Linux) :

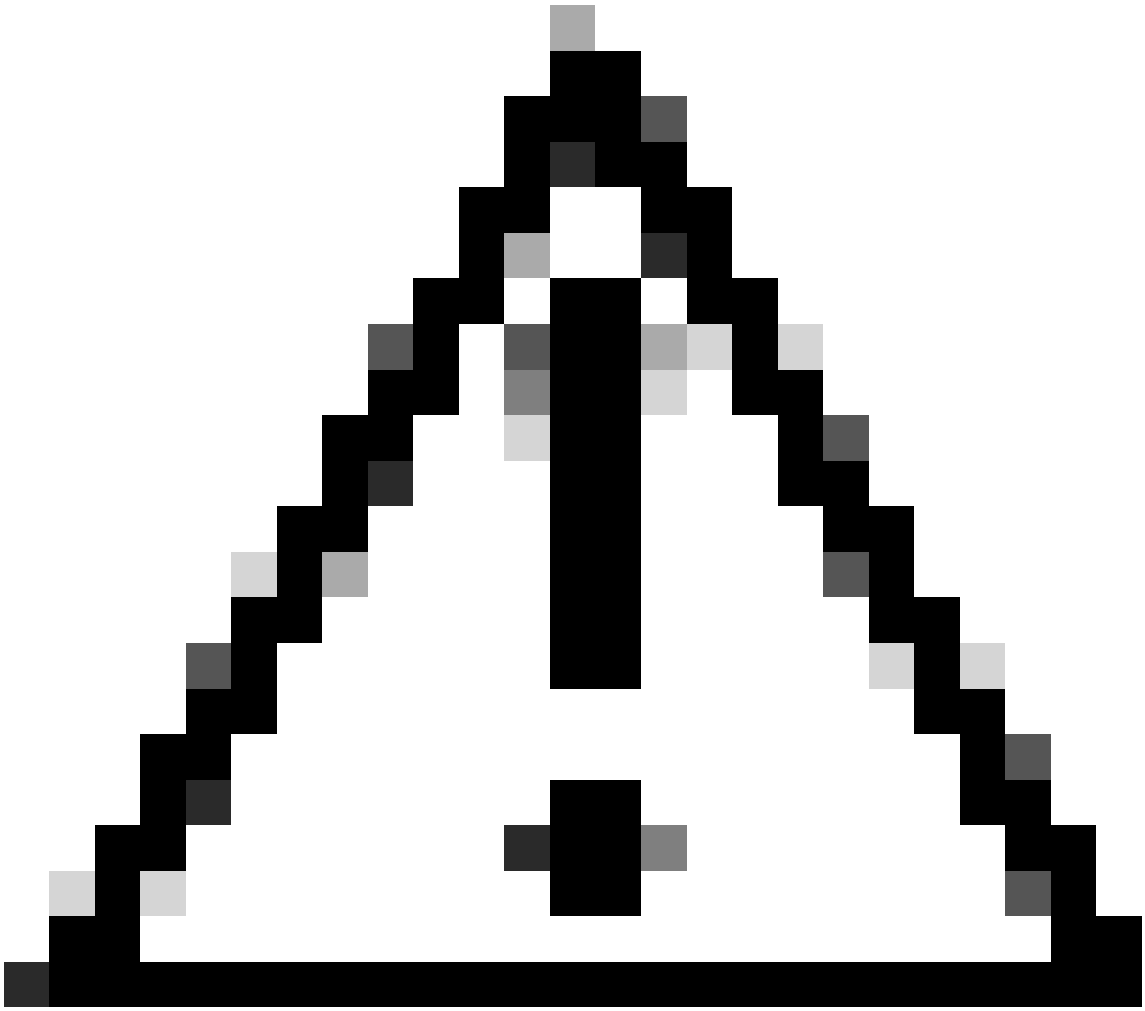
```
RUN yum install -y net-tools iptables
```

    - Pour Ubuntu :

```
RUN apt-get install -y net-tools iptables
```

2. Activation de la journalisation.

```
ENV CSC_LOGGING_OUTPUT=STDOUT
```



Mise en garde : Si cette option est activée, les journaux s'impriment en ligne dans l'interface de ligne de commande, parallèlement aux autres activités en cours.

3. Copiez le package DEB/RPM depuis l'hôte.
   - Pour RHEL :

```
COPY cisco-secure-client-vpn-cli-<VERSION>-1.x86_64.rpm /tmp/cisco-secure-client-cli.rpm
```

   - Pour Ubuntu :

```
COPY cisco-secure-client-vpn-cli_<VERSION>_amd64.deb /tmp/cisco-secure-client-cli.deb
```

4. Afin de démarrer l'agent VPN, de le maintenir en fonctionnement et de le redémarrer si nécessaire, un fichier nommé entry.sh est ajouté comme point d'entrée pour le conteneur Docker. Ce script doit être copié dans le conteneur pour une utilisation ultérieure.

```
#!/bin/bash

wait_forever() {
  while true; do
    sleep infinity &
    wait $!
  done
}

start_service() {
  if [ -f /opt/cisco/secureclient/bin/vpnagentd ]; then
    echo "Starting VPN agent..."
    while true; do
      /opt/cisco/secureclient/bin/vpnagentd -execv_instance &
      SERVICE_PID=$!
      wait $SERVICE_PID
      echo "VPN agent exited. Restarting..."
      sleep 1
    done
  fi
}

start_service
wait_forever
```

- Pour RHEL et Ubuntu :

```
COPY entry.sh /entry.sh
RUN chmod +x /entry.sh
```

5. Installez le package.
   - Pour RHEL :

```
RUN cd /tmp && \
    dnf install -y ./cisco-secure-client-cli.rpm && \
    rm -rf /tmp/cisco-secure-client-cli.rpm
```

   - Pour Ubuntu :

```
RUN cd /tmp && \
    apt-get install -y ./cisco-secure-client-cli.deb && \
    rm -rf /tmp/cisco-secure-client-cli.deb
```

6. Ajoutez le entry.sh comme point d'entrée au conteneur Docker.

```
ENTRYPOINT ["/entry.sh"]
```

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.