

# Dépannez un conteneur de docker quand il ne peut pas accéder à l'Internet

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Problème](#)

[Solution](#)

## Introduction

Ce document décrit comment dépanner un conteneur de docker derrière un serveur proxy quand il ne peut pas accéder à l'Internet.

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Interface de Linux
- Environnements de virtual machine

### [Composants utilisés](#)

Les informations contenues dans ce document sont basées sur les versions de logiciel suivantes :

- Version 4.x de CloudCenter
- Orchestrator de CloudCenter (CCO)

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

## [Informations générales](#)

Si votre entreprise exige le proxy à l'accès à Internet, vous devez configurer le conteneur de docker.

## Problème

C'est la procédure pour reproduire le problème quand le conteneur de docker ne peut pas atteindre l'Internet.

Quand l'utilisateur de base essaye d'exécuter le core\_installer.bin sur le CCO :

```
[root@localhost tmp]# ./core_installer.bin centos7 vmware cco
```

Cette erreur apparaîtra :

```
[root@localhost tmp]# ./core_installer.bin centos7 vmware cco
Verifying archive integrity... All good.
Uncompressing Core Installer V 4.5.2.....
.....
Installing Module: sysupdate
Installing Module: gateway
Installing Module: ntp
Installing Module: jdk8
Installing Module: tomcat8
Installing Module: gwtomcatapr
Installing Module: gwmongodb
Installing Module: docker
Failed in docker. Check /root/cliqr_modules.log for more info
[root@localhost tmp]#
```

## Solution

Étape 1. Exécutez le fichier core\_installer.bin avec ces arguments pour créer le principal répertoire.

```
[root@localhost]# /core_installer.bin --noexec --keep
```

Étape 2. Naviguez vers le principal répertoire.

```
[root@localhost]# cd core
```

Étape 3. Du principal répertoire, exécutez le script de setup.sh pour installer le docker.

```
[root@localhost core]# /setup.sh centos7 vmware docker
```

Le script échoue avec cette erreur « manquée dans le docker ».

Étape 4. Modifiez le Dockerfile.

```
[root@localhost core]# vi docker/cliqr-container-worker/Dockerfile
```

Étape 5. Ajoutez les informations de serveur proxy dans la section ENV du Dockerfile.

```
ENV JAVA_VERSION 1.7.0
ENV http_proxy http://proxy.company.com
ENV https_proxy https://proxy.company.com
```

Remarque: Remplacez proxy.company.com par l'adresse réelle de serveur proxy.

Étape 6. Créez un répertoire de parasite de systemd pour le service de docker.

```
ENV JAVA_VERSION 1.7.0
ENV http_proxy http://proxy.company.com
ENV https_proxy https://proxy.company.com
```

Étape 7. Créez le fichier du docker http-proxy.conf.

```
ENV JAVA_VERSION 1.7.0
ENV http_proxy http://proxy.company.com
ENV https_proxy https://proxy.company.com
```

Étape 8. Ajoutez les informations de serveur proxy.

```
ENV JAVA_VERSION 1.7.0
ENV http_proxy http://proxy.company.com
ENV https_proxy https://proxy.company.com
```

Étape 9. Si vous avez des registres internes de docker que vous devez entrer en contact avec sans proxying, ajoutez-les dans la variable d'environnement NO\_PROXY :

```
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
```

Remarque: Remplacez proxy.company.com par l'adresse réelle de serveur proxy.

Étape 10. Sauvegardez le fichier de configuration et rechargez le service de docker.

```
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
```

Étape 11. Construisez l'employé à la dernière image avec l'aide de ces commandes.

```
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
```

Étape 12. Redémarrez le service de docker.

```
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
```

Étape 13. Testez si le conteneur de docker est configuré.

```
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
```

Une fois que le conteneur de docker est configuré, vous devrez continuer l'installation de core\_installer.bin (si vous installez CCO).

Étape 14. Modifiez le cliqr\_modules.conf.

```
Environment="HTTP_PROXY=http://proxy.company.com"
Environment="HTTPS_PROXY=https://proxy.company.com"
Environment="NO_PROXY=localhost,127.0.0.1,docker-registry.company.com"
```

Étape 15. Ajoutez le docker à l'extrémité du fichier. Ceci indique au core\_installer.bin que le docker est installé.

```
sysupdate
gateway
ntp
jdk8
```

tomcat8  
gwtomcatapr  
gwmongodb  
**docker**

Étape 16. Réexécutez le core\_installer.bin pour se terminer l'installation.

sysupdate  
gateway  
ntp  
jdk8  
tomcat8  
gwtomcatapr  
gwmongodb  
**docker**