

# Redimensionner les clés RSA SSH par défaut sur les bords SD-WAN de Cisco IOS XE

## Table des matières

---

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Configurer](#)

[Diagramme du réseau](#)

[Configurations](#)

[Vérifier](#)

---

## Introduction

Ce document décrit comment augmenter la longueur des clés RSA SSH par défaut utilisées pour les protocoles sécurisés sur les périphéries SD-WAN de Cisco IOS® XE.

## Conditions préalables

### Exigences

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- Réseau étendu défini par logiciel (SD-WAN) Cisco Catalyst
- Fonctionnement de base des clés SSH et des certificats
- Algorithme RSA

### Composants utilisés

- Périphériques SD-WAN Cisco IOS® XE Catalyst 17.9.4a

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 générales

Secure Shell (SSH) est un protocole réseau qui permet aux utilisateurs d'établir des connexions à distance à des périphériques, même sur un réseau non protégé. Le protocole sécurise les

sessions à l'aide de mécanismes cryptographiques standard basés sur une architecture client-serveur.

RSA est Rivest, Shamir, Adleman : Algorithme de chiffrement (système cryptographique à clé publique) qui utilise deux clés : Clé publique et clé privée, également appelée paire de clés. La clé publique RSA est la clé de cryptage et la clé privée RSA est la clé de décryptage.

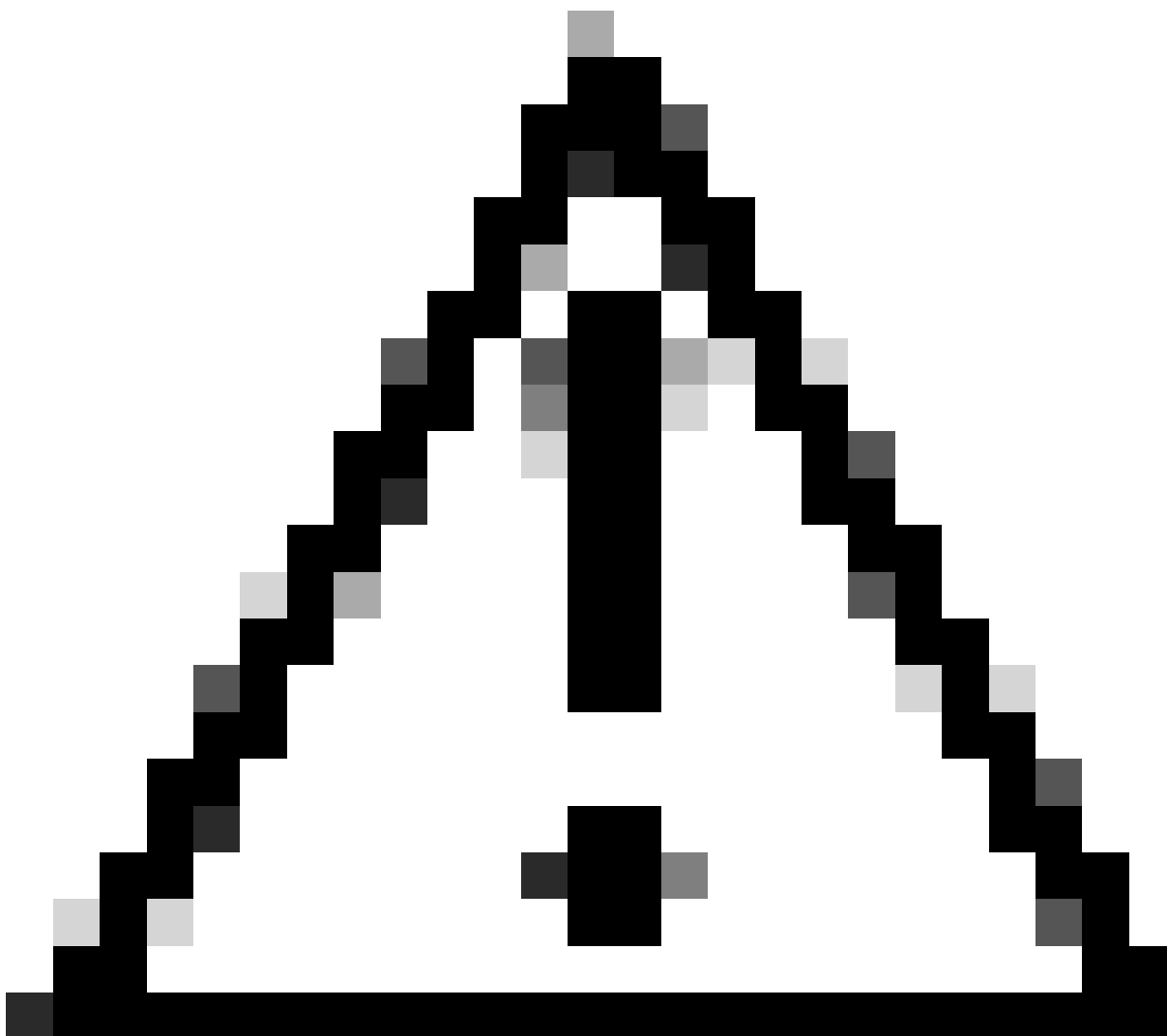
Les clés RSA ont une longueur définie, en bits, du module. Lorsqu'une clé RSA a une longueur de 2048 bits, cela signifie en fait que la valeur du module est comprise entre 22047 et 22048. Les clés publique et privée d'une paire donnée partageant le même module, elles ont également, par définition, la même longueur.

Un certificat trustpoint est un certificat auto-signé, d'où le nom trustpoint, puisqu'il ne dépend pas de la confiance d'une autre personne ou d'une autre partie.

L'infrastructure à clé publique (PKI) Cisco IOS assure la gestion des certificats pour la prise en charge des protocoles de sécurité tels que IP Security (IPSec), Secure Shell (SSH) et Secure Socket Layer (SSL).

Les clés SSH RSA sont importantes sur le SD-WAN de Cisco Catalyst, car elles sont utilisées par le protocole SSH pour établir la communication entre le gestionnaire SD-WAN et les périphériques SD-WAN Edge, car le gestionnaire SD-WAN utilise le protocole Netconf, qui fonctionne sur SSH pour gérer, configurer et surveiller les périphériques.

Pour cette raison, il est nécessaire que les clés soient synchronisées et mises à jour en permanence. Si par conformité et audit, il est nécessaire de modifier la longueur de clé pour la sécurité, il est nécessaire de terminer le processus décrit dans ce document pour redimensionner les clés et les synchroniser correctement sur le certificat pour éviter la déconnexion entre le gestionnaire SD-WAN et les périphériques SD-WAN Edge.



Mise en garde : Veuillez effectuer toutes les étapes du processus pour éviter la perte d'accès au périphérique. Si le périphérique est en production, il est recommandé de l'exécuter dans une fenêtre de maintenance et de disposer d'un accès console au périphérique.

---

## Configurer

Diagramme du réseau

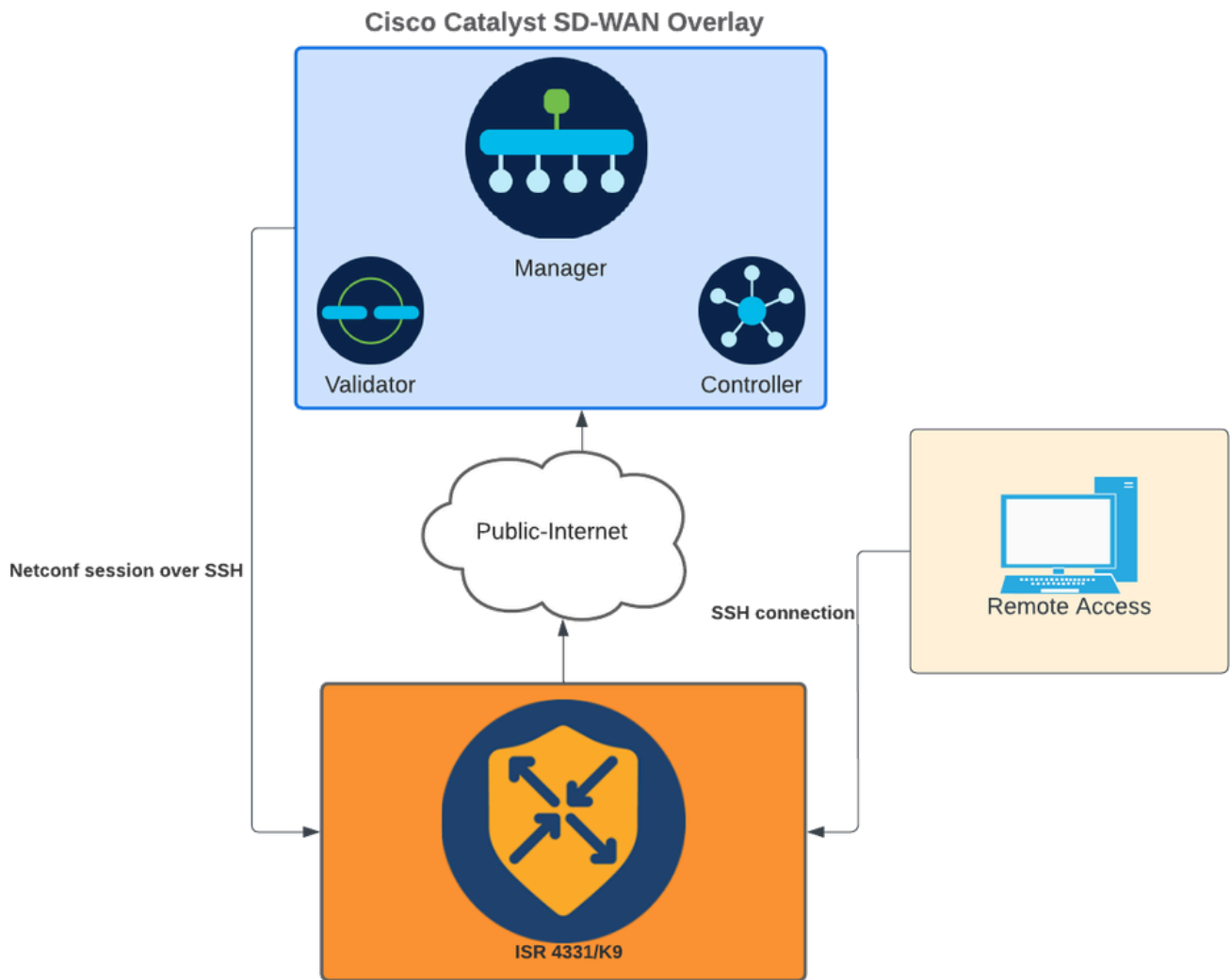


Diagramme du réseau

## Configurations

Les clés RSA dans les périphériques de périphérie WAN ne peuvent être modifiées qu'à l'aide de l'interface de ligne de commande (CLI) ; Les modèles de fonctionnalité complémentaire CLI ne peuvent pas être utilisés pour mettre à jour les clés.



Avertissement : Il est recommandé d'effectuer le processus à l'aide de la console, car l'outil SSH du gestionnaire SD-WAN n'est pas disponible tant que le processus n'est pas terminé.

---



Avertissement : Ce processus nécessite un redémarrage du périphérique. Si le périphérique est en production, il est recommandé de l'exécuter dans une fenêtre de maintenance et de disposer d'un accès console au périphérique. Si aucun accès à la console n'est disponible, configurez temporairement un autre protocole d'accès distant comme telnet.

---

Cet exemple de configuration montre comment supprimer RSA 2048 et utiliser la clé RSA 4096.

1 - Obtient le nom de la clé SSH actuelle.

```
<#root>
```

```
Device#
```

```
show ip ssh
```

```
SSH Enabled - version 2.0
```

```
Authentication methods:publickey,keyboard-interactive,password
```

Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521  
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa  
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr  
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha2-512-etm@openssh.com  
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1  
Authentication timeout: 120 secs; Authentication retries: 3  
Minimum expected Diffie Hellman key size : 2048 bits  
IOS Keys in SECSH format(ssh-rsa, base64 encoded):

TP-self-signed-1072201169 <<<< RSA Key Name

Modulus Size : 2048 bits

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQAZ5urq7f/X+AZJjUnM0dF9pLX+V0jPR8arK6bLSU7d
iGeSDDwW2MPNck/U5HBry9P/L4nKyZ1oEvAhfy7cJVvmoHD41NQW9wb/hLtimuujnRRYkKuIWLmoI7AH
y6YQoetew8XVg1VIjva+JzQ5ZX1JGm8AzN6a95RbRNhGRzgz9cTFmD7m6ArIKZPMYqabXfrY+m/HuQ2
aytbHtJMgm0Qk2fLPak03PnQNYXpiDP3Cm0Eh3LJg82FZQ1eohmhm+mAIInwU4m1LHUouigyBuq1KEBVe
z3vxjB9X8rGF3qzUcx21pHmhXaNpXWen2QQbyfAIDo8WxVoff24uLY1wCVkv
```

2 - Obtenez le certificat autosigné du point de confiance actuel.

<#root>

Device#

show crypto pki trustpoint

Trustpoint TP-self-signed-1072201169: <<<< Self-signed Trustpoint name

Subject Name:

cn=IOS-Self-Signed-Certificate-1072201169

Serial Number (hex): 01

Persistent self-signed certificate trust point

Using key label

TP-self-signed-1072201169

Les deux noms de valeur doivent correspondre.

3 - Supprimez la clé actuelle.

<#root>

Device#

crypto key zeroize rsa

4 - Validez que l'ancienne clé a été supprimée.

```
<#root>  
Device#  
show ip ssh
```

5 - Générez la nouvelle clé.

```
<#root>  
Device#  
crypto key generate rsa modulus 4096 label
```

```
The name for the keys will be: TP-self-signed-1072201169  
% The key modulus size is 4096 bits  
% Generating crypto RSA keys in background ...  
*Jun 25 21:35:18.919: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-1072201169 has been generated  
*Jun 25 21:35:18.924: %SSH-5-ENABLED: SSH 2.0 has been enabled  
*Jun 25 21:35:23.205: %CRYPTO_ENGINE-5-KEY_ADDITION: A key named TP-self-signed-1072201169 has been generated  
*Jun 25 21:35:29.674: %SYS-6-PRIVCFG_ENCRYPT_SUCCESS: Successfully encrypted private config file
```

Ce processus peut prendre de 2 à 5 minutes.

6 - Validez la nouvelle clé générée.

```
<#root>  
Device#  
show ip ssh
```

```
SSH Enabled - version 2.0  
Authentication methods:publickey,keyboard-interactive,password  
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521  
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa  
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr  
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,
```



KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1  
Authentication timeout: 120 secs; Authentication retries: 3  
Minimum expected Diffie Hellman key size : 2048 bits

IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1072201169

Modulus Size : 4096 bits <<<< Key Size

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQDE0t/SX3oQKN6z0Wv0aFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQOGGsdxo2+2Y/idAFm808mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrDzpJ6d8RgUWLOtgHSzQ7P796c0B1YLtK3eF00H1AFmFy5ec8Own7ik0
JjKtwEozImFMjHZfUEUjFuhPJELB06yYEipPwMRaZYFfTRbNjM8/7S0JG1FkgFVW5nITTIgISoMV8EJv
bL18cVgATDb10ckeDb7uU6PDXm3zonmZC0yqHtF10A0JxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5et0049UH0y0bMgPuWrJpwtk1iYA3+t6N/Qd1C5VSoua+TsMfp7Dh3k6qUTFUSy2h3
Kiibov1HKyvkccQxi6nDfAKb8o+Z8/43xbvW1DIKAuj1rbdyqPAJB411TZJk0Hk8zRP5gZ8u4jtjNKQHb
vNa3ieg4RLED0x41qCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MMOu14YDJYr
D+jnyoZr4PnfwAgk4M9U89deWS1IRPMIXYd35YmLvD60eQ5EQALNiNPUEkpdPKs4orYysEV0pRoY+HQ
```

Une nouvelle clé est générée. Cependant, au moment où l'ancienne clé a été supprimée, le certificat auto-signé utilisé par les sessions Netconf est également supprimé du point de confiance.

<#root>

Device#

sh crypto pki trustpoint status


```
Trustpoint TP-self-signed-1072201169:
Issuing CA certificate configured::
Issuing CA certificate configured:
Subject Name:
cn=Cisco Licensing Root CA,o=Cisco
Fingerprint MD5: 1468DC18 250BDFCF 769C29DF E1F7E5A8
Fingerprint SHA1: 5CA95FB6 E2980EC1 5AFB681B BB7E62B5 AD3FA8B8
State:
```

Keys generated ..... No <<<< Depending on the version, it can erase the key or even that, delete

```
Issuing CA authenticated ..... Yes
Certificate request(s) ..... None
```

Une fois la nouvelle clé 4096 générée, les clés ne sont pas automatiquement mises à jour sur le certificat auto-signé, et il est nécessaire d'effectuer des étapes supplémentaires pour la mettre à jour.

---

 Remarque : Si seule la clé est générée, mais n'est pas mise à jour dans le certificat, le gestionnaire SD-WAN perd les sessions Netconf, ce qui peut interrompre toutes les activités de gestion du périphérique (modèles, configuration, etc.).

---

Il existe deux façons de générer le certificat/attribuer la clé :

1 - Rechargez le périphérique.

```
<#root>
```

```
Device#
```

```
reload
```

2 - Redémarrez HTTP secure-server. Cette option n'est disponible que si le périphérique est en mode CLI.

```
<#root>
```

```
Device (config)#
```

```
no ip http secure-server
```

```
Device (config)#
```

```
commit
```

```
Device (config)#
```

```
ip http secure-server
```

```
Device (config)#
```

```
commit
```

## Vérifier

Après le rechargement, vérifiez que la nouvelle clé est générée et que le certificat est sous le point de confiance portant le même nom.

```
<#root>
```

```
Device#
```

```
show ip ssh
```

```
SSH Enabled - version 2.0
```

```
Authentication methods:publickey,keyboard-interactive,password
```

```
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521
```

```
Hostkey Algorithms:x509v3-ssh-rsa,rsa-sha2-512,rsa-sha2-256,ssh-rsa
```

```
Encryption Algorithms:aes128-gcm,aes256-gcm,aes128-ctr,aes192-ctr,aes256-ctr
```

```
MAC Algorithms:hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
```

```
KEX Algorithms:ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1
```

Authentication timeout: 120 secs; Authentication retries: 3  
Minimum expected Diffie Hellman key size : 2048 bits

IOS Keys in SECSH format(ssh-rsa, base64 encoded): TP-self-signed-1072201169

Modulus Size : 4096 bits

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDE0t/SX3oQKN6z0Wv0aFAkMcaZNzQ6JgP+7xjuX143
YS7YGmOPwIPgs8N2LWvmdLXQ/PqsQOGGsdxo2+2Y/idAFm808mb6bcWFU+t3b/Pf6GBzUv8SPnR4i4nN
5GYhZE9HX3REWYp7d+7l1YawrDzpJ6d8RgUWLOtgHSzQ7P796c0B1YLtK3eF00H1AFmFy5ec8Own7ik0
JjKtwEozImFMjHZfUEUjFuhPJELB06yYEipPWMRaZYFfTRbNjM8/7S0JG1FkgFVW5nITTIgISoMV8EJv
bL18cVgATDb10ckeDb7uU6PDxm3zonmZC0yqHtF10A0JxUpUa6Iry1XwMzzZqDdu32F5If4/SSCmbHV2
46P8AjCdu/2TKK5et0049UH0y0bMgPuWrJpwtk1iYA3+t6N/Qd1C5VSoua+TsMfp7Dh3k6qUTFUSy2h3
Kiibov1HKYvkccqXi6nDfAKb8o+Z8/43xbvW1DIKAuj1rbdyqPAJB411TZJk0Hk8zRP5gZ8u4jTjNKQHb
vNa3ieg4RLED0x41qCk+iSRzdddMq2te1xSWFPh67i4BnJHvhVnR6LF5Gu+uF5TWwcpy2MM0u14YDJYr
D+jnyoZr4PnfwAgk4M9U89deWS1IRPMIXYd35YmLvD60eQ5EQALNiNPUEkpdPKs4orYysEV0pRoY+HQ
```

<#root>

Device#

show crypto pki trustpoint

Trustpoint TP-self-signed-1072201169: <<<< Trustpoint name

Subject Name:

cn=IOS-Self-Signed-Certificate-1072201169

Serial Number (hex): 01

Persistent self-signed certificate trust point

Using key label TP-self-signed-107220116

<#root>

Device#

show crypto pki certificates

Router Self-Signed Certificate

Status: Available

Certificate Serial Number (hex): 01

Certificate Usage: General Purpose

Issuer:

cn=IOS-Self-Signed-Certificate-1072201169

Subject:

Name: IOS-Self-Signed-Certificate-1072201169

cn=IOS-Self-Signed-Certificate-1072201169

Validity Date:

start date: 21:07:33 UTC Dec 27 2023

end date: 21:07:33 UTC Dec 26 2033

Associated Trustpoints: TP-self-signed-1072201169

Storage: nvram:IOS-Self-Sig#4.cer

Vérifiez que SD-WAN Manager peut appliquer les modifications de configuration au routeur du périphérique.

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.