

# Améliorez une application utilisant CloudCenter

## Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Définissez le processus de mise à niveau](#)

[Créez la nouvelle version](#)

[Déployez l'application](#)

## Introduction

Ce document décrit le processus pour améliorer une application utilisant CloudCenter.

## Conditions préalables

### Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- CloudCenter
- Coup

### [Composants utilisés](#)

Les informations dans ce document sont basées sur CloudCenter 4.8.1.1.

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](#)

Il y a de plusieurs manières d'améliorer une application dans CloudCenter. Une option est de créer une action faite sur commande qui peut être appliquée à différentes VMs et exécute un script de mise à jour. Cette méthode te donne le contrôle complet au-dessus de la mise à jour et permet le test d'un noeud avant d'améliorer le noeud suivant. Le du côté incliné est que c'est un processus très manuel qui exige les scripts individualisés par écriture pour chaque mise à jour. La méthode préférée est de se servir du cadre de la mise à jour de CloudCenter pour automatiser le processus de mise à niveau.

# Définissez le processus de mise à niveau

Edit "Upgrade Application" Application Profile

Version: [1.0](#) (Revision: 3) > [2.0](#)

Basic Information Global Parameters **Topology Modeler**

Services

- Message Bus
- OS Service
- Custom Service
- File System
- Workflow
- Orchestration
- Frontend Cache
- Load Balancer
- Web Server

Apache2  
Open-source HTTP server for OS

Geronomo3  
Open source application server

IIS  
Web server for Windows-based apps

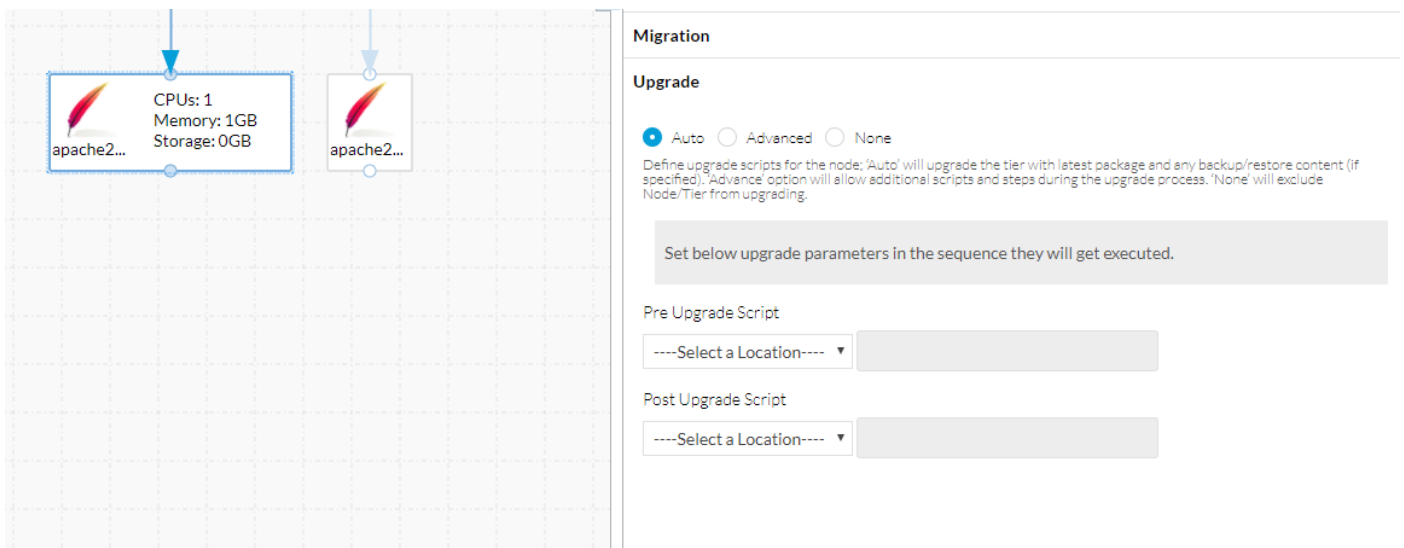
Jetty

```
graph TD; nginx_1[nginx_1] --> apache2_1[apache2...]; nginx_1 --> apache2_2[apache2...];
```

The diagram illustrates a topology modeler interface. On the left, there is a sidebar with a list of services. Under the 'Web Server' category, several options are visible: Apache2 (Open-source HTTP server for OS), Geronomo3 (Open source application server), IIS (Web server for Windows-based apps), and Jetty. The main workspace shows a network diagram on a grid. At the top, there is a box labeled 'nginx\_1' with the NGINX logo. Two lines connect this box to two separate boxes below, each labeled 'apache2...' with the Apache logo. The interface includes search, zoom, and refresh icons at the top left, and a 'Clear' button at the top right.

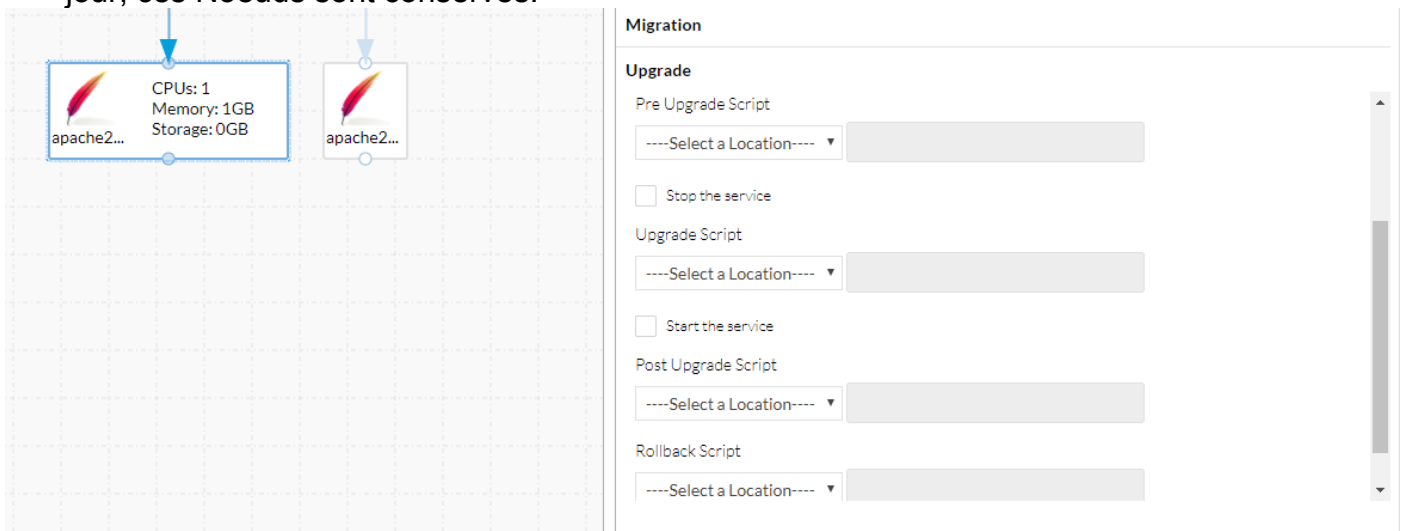
Dans cet exemple d'applications, il y a deux web server d'Apache derrière un équilibreur de charge de Nginx. Ces web server sont identiques et fournissent la Disponibilité ha à un site Web qui est hébergé. Un processus de mise à niveau idéal permet les Noeuds à mettre à jour individuellement de sorte qu'il y ait toujours un noeud accueillant le site Web tenant compte de la disponibilité de 100% pendant le processus de mise à niveau.

Par défaut, pendant une mise à jour CloudCenter télécharge n'importe quels nouveaux modules et contenu, puis se sert de tous les scripts de sauvegarde et de restauration pour persister des données. Si une logique plus en profondeur est nécessaire, alors des scripts de mise à jour peuvent être inclus.



Sous l'onglet de **transfert**, les scripts de sauvegarde et de restauration peuvent être trouvés. Ceux sont utilisés pour le transfert et la mise à jour. L'onglet de **mise à jour** a trois options : **Automatique, avancé, aucun**.

- L'automatique permet à CloudCenter pour améliorer automatiquement le noeud, il télécharge le nouveau contenu, et exécute les scripts de sauvegarde et de restaurations pour préserver les informations importantes.
- Avancé permet le contrôle complet du processus de mise à niveau.
- Aucun ne signifie n'améliore pas ce noeud, il peut être fait pour les Noeuds qui n'ont aucune modification de version à la version, telle qu'un équilibreur de charge. Pendant une mise à jour, ces Noeuds sont conservés.



Avancé laisse plus de scripts à ajouter et te permet pour arrêter et commencer le service pendant la mise à jour.

Une fois que toutes les actions nécessaires de mise à jour sont définies, il est important de **sauvegarder** l'application avant de passer à l'étape suivante

## Créez la nouvelle version

Après que vous sauvegardiez l'application, naviguez de nouveau au **modeleur de topologie**.

Edit "Upgrade Application" Application Profile

Version: [1.0](#) (Revision: 3) > [2.0](#)

Basic Information Global Parameters **Topology Modeler**

Services

- Message Bus
- OS Service
- Custom Service
- File System
- Workflow
- Orchestration
- Frontend Cache
- Load Balancer
- Web Server

Apache2  
Open-source HTTP server for OS

Geronomo3  
Open source application server

IIS  
Web server for Windows-based apps

Jetty  
Java-based HTTP...

```
graph TD; nginx_1[nginx_1] --> apache2_1[apache2...]; nginx_1 --> apache2_2[apache2...];
```

Traitements de CloudCenter améliorant avec l'aide de versioning. L'application dans l'image ci-dessus est à la version 1.0, ceci peut être vue dans le coin supérieur gauche. Afin de se servir de l'outil de la mise à jour de CloudCenter, une nouvelle version doit être faite.

- **Les informations de base** choisies.
- Écrivez une nouvelle **version**.

# Edit "Upgrade Application" Application Profile

Version: [1.0](#) (Revision: 3) > [2.0](#)

Basic Information

Global Parameters

Topology Modeler

Web App Name \*

Upgrade Application

Version \*

2.0

Revision

3

CloudCenter enregistre la version 1.0 et met tous les nouveaux changements de version 2.0.

Ceci indique à CloudCenter qu'il y a une nouvelle version, et lui permet de dépister les différences. Puisque cette application est juste deux web server, la seule différence est de mettre à jour le **progiciel** pour indiquer un nouveau fichier zip.

L'application peut être enregistrée de nouveau.

## Déployez l'application

Maintenant, quand vous déployez l'application, vous pouvez choisir la version à se déployer. Pour cet exemple, la version originale est déployée.

# General Settings

## \* DEPLOYMENT NAME

UpgradeExample

## \* APPLICATION VERSION

2.0

1.0

2.0

Enter Tag Name

## TERMINATE PROTECTION



## AGING POLICY

Une fois que l'application est déployée elle peut être mise à jour de l'écran de déploiements.

Application	Status	Environment	Deployed At	Duration	Cost	Actions
UpgradeExample Upgrade Application (V1.0) AWS/us-east-1	Deployed	Dev	20 Dec 2017 at 08:56 AM	6 mins	\$0.04	-Actions- Suspend Terminate Terminate And Hide <b>Upgrade</b> Promote Migrate Enable Terminate Protection Share
queueManTest8 QueueMan (V2.0) AWS/us-east-1	Stopping	Dev	19 Dec 2017 at 02:33 PM	18 hrs 25 mins	\$0.23	
QueueManTest6 QueueMan (V2.0) AWS/us-east-1	Terminating	Dev	19 Dec 2017 at 02:05 PM	18 hrs 53 mins	\$0.23	
QueueManTest5 QueueMan (V2.0) AWS/us-east-1	Terminated	Dev	19 Dec 2017 at 01:55 PM	9 mins	\$0.01	
QueueManTest3						

Le processus de mise à niveau commence à partir du plus bas niveau et se produit un noeud à la fois. Pour notre application à deux étages, un Apache Web Server est mis à jour.

Une fois que cela est terminé, le deuxième est mis à jour. Si vous avez défini un processus de mise à niveau pour l'équilibreur de charge de Nginx, il est mis à jour dans durent.