

Aktualisieren einer Anwendung mit CloudCenter

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Upgrade-Prozess definieren](#)

[Neue Version erstellen](#)

[Anwendung bereitstellen](#)

Einführung

Dieses Dokument beschreibt den Prozess zum Aktualisieren einer Anwendung mit CloudCenter.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- CloudCenter
- Bash

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf CloudCenter 4.8.1.1.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundinformationen

Es gibt mehrere Möglichkeiten, eine Anwendung in CloudCenter zu aktualisieren. Eine Möglichkeit besteht darin, eine benutzerdefinierte Aktion zu erstellen, die auf einzelne VMs angewendet werden kann und ein Upgrade-Skript ausführt. Diese Methode bietet Ihnen vollständige Kontrolle über die Aktualisierung und ermöglicht das Testen eines Knotens, bevor Sie den nächsten Knoten aktualisieren. Der Nachteil ist, dass es ein sehr manueller Prozess ist, der das Schreiben individueller Skripte für jedes Upgrade erfordert. Die bevorzugte Methode ist die Nutzung des CloudCenter-Upgrade-Frameworks zur Automatisierung des Upgrade-Prozesses.

Upgrade-Prozess definieren

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

Geronimo3
Open source application server

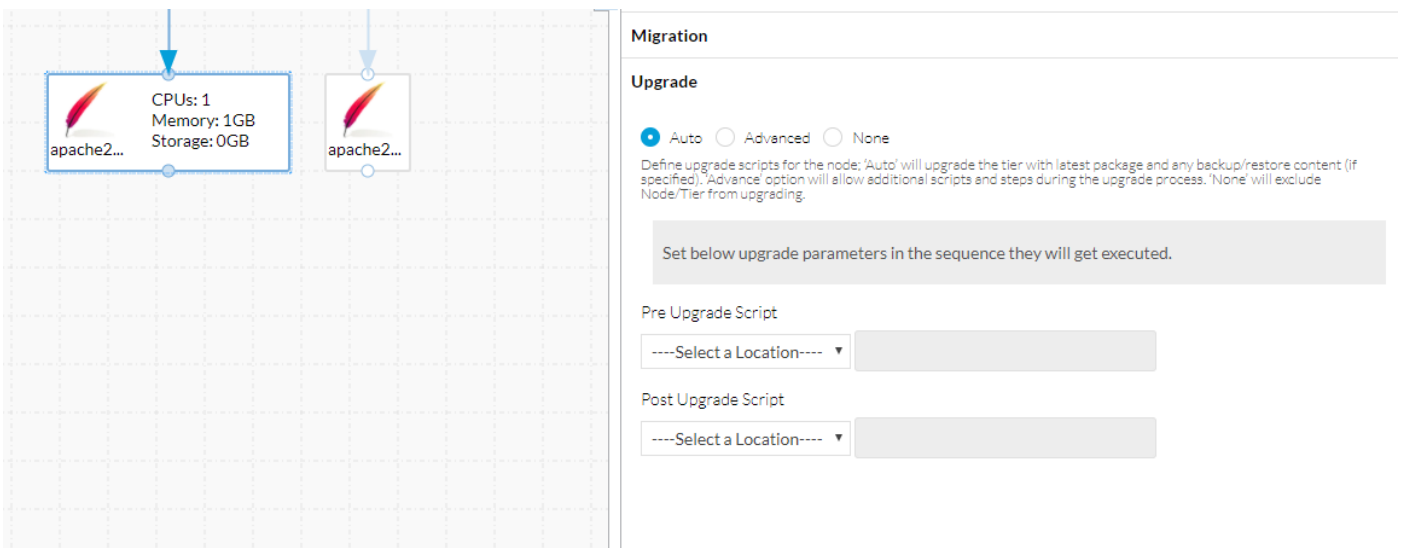
IIS
Web server for Windows-based apps

Jetty
Java based HTTP server

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

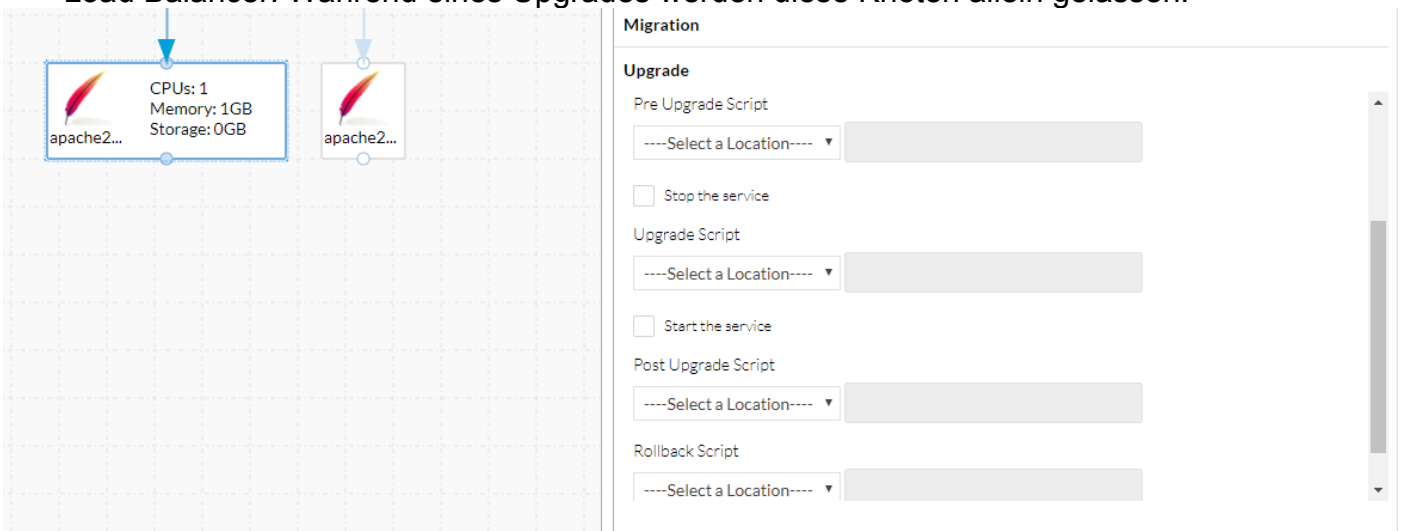
In dieser Beispielanwendung befinden sich hinter einem Nginx Load Balancer zwei Apache-Webserver. Diese Webserver sind identisch und stellen eine HA-Verfügbarkeit für eine gehostete Website bereit. Ein idealer Upgrade-Prozess ermöglicht es, die Knoten einzeln zu aktualisieren, sodass immer ein Knoten die Website hostet, der eine 100%ige Verfügbarkeit während des Upgrade-Prozesses ermöglicht.

Standardmäßig lädt CloudCenter während eines Upgrades alle neuen Pakete und Inhalte herunter und verwendet dann alle Backup- und Wiederherstellungs-Skripts, um Daten zu erhalten. Wenn eine detailliertere Logik erforderlich ist, können Upgrade-Skripts eingeschlossen werden.



Auf der Registerkarte **Migration** finden Sie die Sicherungs- und Wiederherstellungs-Skripts. Diese werden sowohl für die Migration als auch für Upgrades verwendet. Die Registerkarte **Upgrade** bietet drei Optionen: **Auto**, **Advanced**, **None**.

- Auto (Automatisch) ermöglicht CloudCenter die automatische Aktualisierung des Knotens, lädt den neuen Inhalt herunter und führt die Backups aus und stellt Skripts wieder her, um wichtige Informationen zu erhalten.
- Advanced ermöglicht die vollständige Kontrolle des Upgrade-Prozesses.
- Kein Hinweis bedeutet, dass dieser Knoten nicht aktualisiert wird. Dies kann für Knoten durchgeführt werden, die keine Änderungen von Version zu Version aufweisen, z. B. einen Load Balancer. Während eines Upgrades werden diese Knoten allein gelassen.



Advanced ermöglicht das Hinzufügen weiterer Skripts und das Anhalten und Starten des Service während des Upgrades.

Nachdem alle erforderlichen Upgrade-Aktionen definiert sind, ist es wichtig, die Anwendung zu **speichern**, bevor Sie mit dem nächsten Schritt fortfahren.

Neue Version erstellen

Navigieren Sie nach dem Speichern der Anwendung zurück zum **Topologiemanager**.

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...

The diagram shows a central node labeled 'nginx_1' with the NGINX logo. Two lines extend downwards from this node, each ending in an arrowhead pointing to a node labeled 'apache2...'. Each 'apache2...' node features the Apache logo. The entire diagram is set against a light gray grid background. At the top of the diagram area, there are search and refresh icons, and a 'Clear' button.

CloudCenter übernimmt das Upgrade mithilfe der Versionsverwaltung. Die Anwendung im Bild oben befindet sich in Version 1.0, dies ist in der oberen linken Ecke zu sehen. Um das Upgrade-Tool von CloudCenter nutzen zu können, muss eine neue Version erstellt werden.

- Wählen Sie **Basisinformationen** aus.
- Geben Sie eine neue **Version** ein.

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 speichert Version 1.0 und fügt alle neuen Änderungen in Version 2.0 ein.

Dies weist CloudCenter auf eine neue Version hin und ermöglicht es, die Unterschiede zu verfolgen. Da es sich bei dieser Anwendung nur um zwei Webserver handelt, besteht der einzige Unterschied darin, das **Anwendungspaket** auf eine neue ZIP-Datei zu aktualisieren.

Die Anwendung kann erneut gespeichert werden.

Anwendung bereitstellen

Wenn Sie die Anwendung bereitstellen, können Sie jetzt auswählen, welche Version bereitgestellt werden soll. In diesem Beispiel wird die Originalversion bereitgestellt.

General Settings

* DEPLOYMENT NAME

UpgradeExample

* APPLICATION VERSION

2.0

1.0

2.0

Enter Tag Name

TERMINATE PROTECTION



AGING POLICY

Nachdem die Anwendung bereitgestellt wurde, kann sie über den Bildschirm "Bereitstellungen" aktualisiert werden.

Icon	Name	Status	Environment	Created 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- -Actions- Suspend Terminate Terminate And Hide Upgrade Promote Migrate Enable Terminate Protection Share
	queueManTest8 QueueMan (V2.0) AWS/us-east-1	Stopping	Dev	19 Dec 2017 at 02:39 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						

Der Upgrade-Prozess beginnt mit der niedrigsten Ebene und findet jeweils nur einen Knoten statt. Für unsere zweistufige Anwendung wird ein Apache-Webserver aktualisiert.

Nach Abschluss dieses Vorgangs wird das zweite aktualisiert. Wenn Sie einen Aktualisierungsvorgang für den Nginx Load Balancer definiert haben, wird dieser im letzten Schritt aktualisiert.