

Actualice una aplicación usando CloudCenter

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Antecedentes](#)

[Defina el proceso de actualización](#)

[Cree la nueva versión](#)

[Despliegue la aplicación](#)

Introducción

Este documento describe el proceso para actualizar una aplicación usando CloudCenter.

Prerequisites

Requisitos

Cisco recomienda que tenga conocimiento sobre estos temas:

- CloudCenter
- Golpe

Componentes Utilizados

La información en este documento se basa en CloudCenter 4.8.1.1.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

Antecedentes

Hay diferentes formas de actualizar una aplicación en CloudCenter. Una opción es crear una acción personalizada que se pueda aplicar a los VM individuales y funcione con una secuencia de comandos de actualización. Este método le da el control total sobre la actualización y permite la prueba de un nodo antes de actualizar el siguiente nodo. La desventaja es que es mismo un Proceso manual que requiere los scripts individualizados escritura para cada actualización. El método preferido es hacer uso del marco de la actualización de CloudCenter para automatizar el proceso de actualización.

Defina el proceso de actualización

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

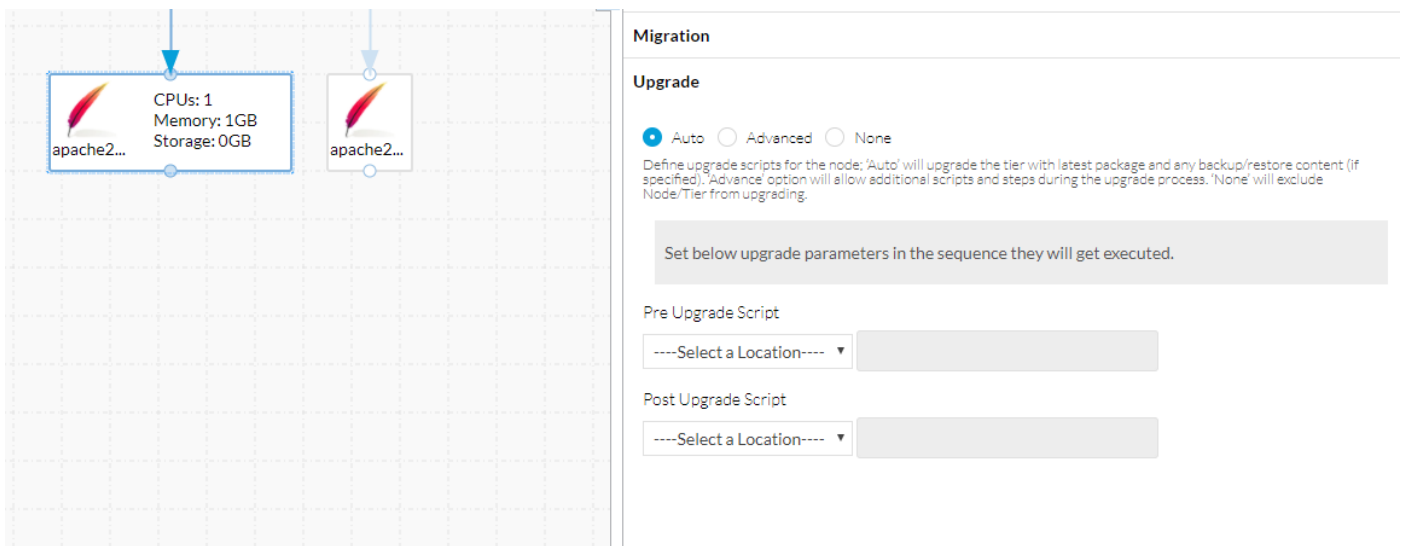
IIS
Web server for Windows-based apps

Jetty
Java-based HTTP server

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

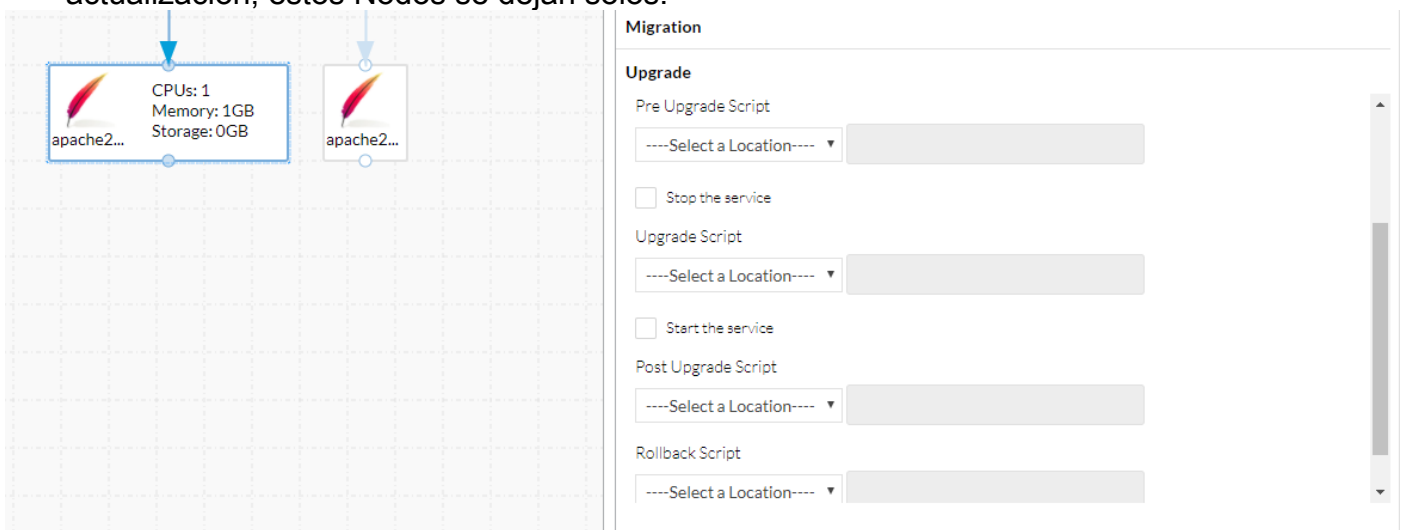
En esta aplicación de ejemplo, hay dos servidores Web de Apache detrás de un balanceador de la carga de Nginx. Estos servidores Web son idénticos y proporcionan la Disponibilidad HA a un sitio web se esté recibiendo que. Un proceso de actualización ideal permite que los Nodos sean actualizados individualmente de modo que haya siempre un nodo que recibe el sitio web teniendo en cuenta el uptime del 100% durante el proceso de actualización.

Por abandono, durante una actualización CloudCenter descarga cualesquiera nuevos paquetes y contenido, después hace uso de cualesquiera scripts de reserva y del restore para persistir los datos. Si una lógica más profundizada es necesaria, después las secuencias de comandos de actualización pueden ser incluidas.



Bajo lengüeta de la **migración**, los scripts de reserva y del restore pueden ser encontrados. Ésos se utilizan para la migración y la actualización. La lengüeta de la **actualización** tiene tres opciones: **Auto**, **avanzado**, **ningunos**.

- El auto permite que CloudCenter actualice automáticamente el nodo, descarga el nuevo contenido, y funciona con los scripts de reserva y de los restores para preservar la información importante.
- Avanzado permite el control total del proceso de actualización.
- Ninguno significa no actualiza este nodo, él se puede hacer para los Nodos que no tienen ningún cambio de la versión a la versión, tal como un balanceador de la carga. Durante una actualización, estos Nodos se dejan solos.



Avanzado permite más scripts sean agregados y permite que usted pare y que comience el servicio durante la actualización.

Una vez que se definen todas las acciones necesarias de la actualización, es importante **salvar la aplicación** antes de mover encendido al siguiente paso

Cree la nueva versión

Después de que usted salve la aplicación, navegue de nuevo al **modelador de la topología**.

The screenshot shows the 'Edit "Upgrade Application" Application Profile' interface. At the top, it indicates the current version is 1.0 (Revision: 3) and a new version 2.0 is being created. The 'Topology Modeler' tab is active, showing a diagram where an NGINX node (nginx_1) is connected to two Apache2 nodes. The left sidebar lists various services, including Apache2, Geronimo3, IIS, and Jetty.

Manijas de CloudCenter que actualizan con la ayuda de versioning. La aplicación en la imagen arriba está en la versión 1.0, esto se puede considerar en la esquina superior izquierdo. Para hacer uso de la herramienta de actualización de CloudCenter, una nueva versión debe ser hecha.

- Seleccione la **información básica**.
- Ingrese una nueva versión.

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 guarda la versión 1.0 y pone todos los nuevos cambios en la versión 2.0.

Esto dice a CloudCenter que hay una nueva versión, y permite que siga las diferencias. Puesto que esta aplicación es apenas dos servidores Web, la única diferencia es poner al día el **paquete de la aplicación** para señalar a un nuevo archivo zip.

La aplicación se puede guardar otra vez.

Despliegue la aplicación

Ahora, cuando usted despliega la aplicación, usted puede elegir que versión a desplegar. Por este ejemplo, se despliega la versión original.

General Settings

* DEPLOYMENT NAME

UpgradeExample

* APPLICATION VERSION

2.0

1.0

2.0

Enter Tag Name

TERMINATE PROTECTION



AGING POLICY

Una vez que se despliega la aplicación puede ser actualizada de la pantalla de las implementaciones.

Application Name	State	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- -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						

El proceso de actualización empieza con la grada más baja y sucede un en un momento del nodo. Para nuestra aplicación de dos niveles, uno servidor Web Apache se actualiza.

Una vez que se completa eso, se actualiza el segundo. Si usted ha definido un proceso de actualización para el balanceador de la carga de Nginx, se actualiza en dura.