

Promova um aplicativo usando CloudCenter

Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Defina o processo de upgrade](#)

[Crie a nova versão](#)

[Distribua o aplicativo](#)

Introdução

Este documento descreve o processo para promover um aplicativo usando CloudCenter.

Pré-requisitos

Requisitos

A Cisco recomenda que você tenha conhecimento destes tópicos:

- CloudCenter
- Festança

Componentes Utilizados

A informação neste documento é baseada em CloudCenter 4.8.1.1.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

Informações de Apoio

Há umas formas múltiplas promover um aplicativo em CloudCenter. Uma opção é criar uma ação feita sob encomenda que possa ser aplicada aos VM individuais e execute um script de upgrade. Este método dá-lhe o controle completo sobre a elevação e permite-o um teste de um nó antes de promover o próximo nó. O downside é que é muito um processo manual que exija scripts particularizados escrita para cada elevação. O método preferido é utilizar a estrutura da elevação de CloudCenter para automatizar o processo de upgrade.

Defina o processo de upgrade

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

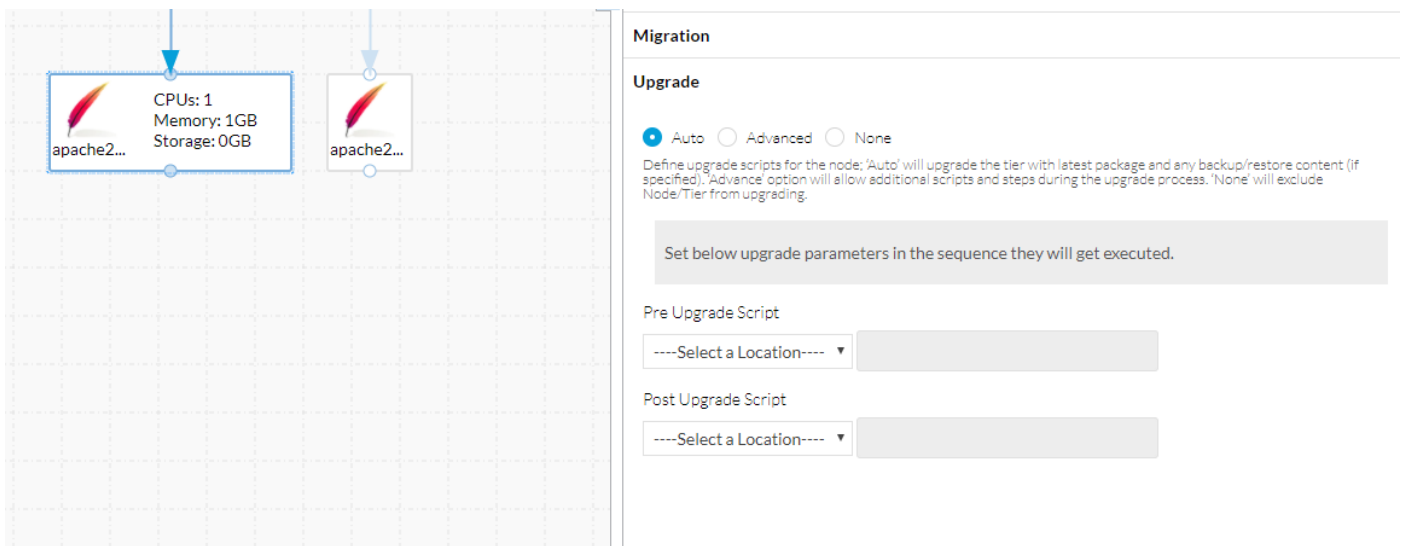
Jetty
Java-based HTTP server

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

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), Geronimo3 (Open source application server), IIS (Web server for Windows-based apps), and Jetty (Java-based HTTP server). The main area of the interface shows a network diagram on a grid background. At the top center, there is a box labeled 'nginx_1' with the NGINX logo. Two lines connect this box to two separate boxes below it, both labeled 'apache2...'. Each box contains the Apache logo. The interface also includes search, zoom, and refresh icons at the top left, and a 'Clear' button at the top right.

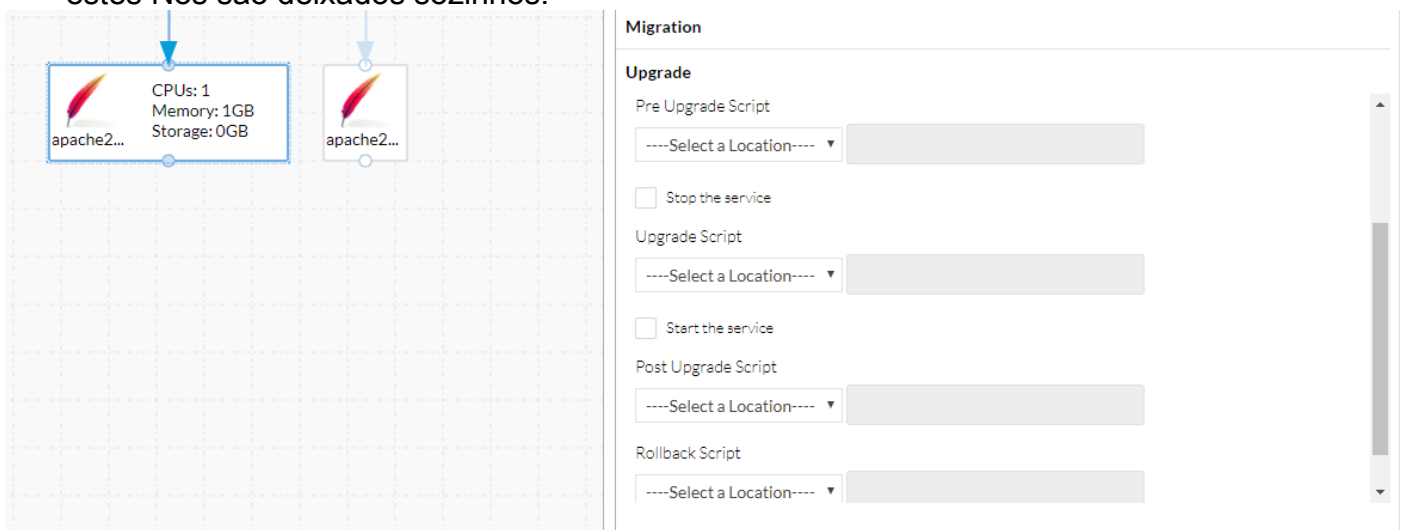
Neste aplicativo de amostra, há dois servidores da Web Apache atrás de um equilibrador da carga de Nginx. Estes servidores de Web são idênticos e fornecem a Disponibilidade HA a um Web site que esteja sendo hospedado. Um processo de upgrade ideal permite que os Nós sejam promovidos individualmente de modo que haja sempre um nó que hospeda o Web site permitindo o uptime de 100% durante o processo de upgrade.

À revelia, durante uma elevação CloudCenter transfere todos os pacotes e índice novos, a seguir utiliza quaisquer scripts alternativos e da restauração para persistir dados. Se uma lógica mais detalhada é precisada, a seguir os script de upgrade podem ser incluídos.



Sob a aba da **migração**, os scripts alternativos e da restauração podem ser encontrados. Aqueles são usados para a migração e a elevação. A aba da **elevação** tem três opções: **Auto**, **avançado**, **nenhuns**.

- O automático permite que CloudCenter promova automaticamente o nó, transfere o índice novo, e executa os scripts alternativos e das restaurações para preservar a informação importante.
- Avançado permite o controle completo do processo de upgrade.
- Nenhum significa não promove este nó, ele pode ser feito para os Nós que não têm nenhuma mudança da versão à versão, tal como um equilibrador da carga. Durante uma elevação, estes Nós são deixados sozinhos.



Avançado permite mais scripts sejam adicionados e permite que você pare e comece o serviço durante a elevação.

Uma vez que todas as ações necessárias da elevação são definidas, é importante **salvar o** aplicativo antes de mover-se sobre para a próxima etapa

Crie a nova versão

Depois que você salvar o aplicativo, navegue de volta ao **modelador da topologia**.

The screenshot displays the 'Edit "Upgrade Application" Application Profile' window. At the top, it shows the version information: 'Version: 1.0 (Revision: 3) > 2.0'. Below this, there are three tabs: 'Basic Information', 'Global Parameters', and 'Topology Modeler'. The 'Topology Modeler' tab is active, showing a diagram of a web application architecture. The diagram consists of a central 'nginx_1' node at the top, which is connected to two 'apache2...' nodes below it. On the left side, there is a 'Services' panel with a list of service categories: Message Bus, OS Service, Custom Service, File System, Workflow, Orchestration, Frontend Cache, Load Balancer, and Web Server. Under the 'Web Server' category, several options are listed: Apache2 (Open-source HTTP server for OS), Geronimo3 (Open source application server), IIS (Web server for Windows-based apps), and Jetty (Java-based HTTP server).

Punhos de CloudCenter que promovem com a ajuda de versioning. O aplicativo na imagem acima está na versão 1.0, isto pode ser considerado no canto esquerdo superior. A fim utilizar a ferramenta de upgrade de CloudCenter, uma nova versão deve ser feita.

- Selecione a **informação básica**.
- Incorpore uma nova versão.

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 salvar a versão 1.0 e põe todas as mudanças novas na versão 2.0.

Isto diz a CloudCenter que há uma nova versão, e permite que siga as diferenças. Desde que este aplicativo é apenas dois servidores de Web, a única diferença é atualizar o **pacote de aplicativo** para apontar a um arquivo zip novo.

O aplicativo pode ser salvar outra vez.

Distribua o aplicativo

Agora, quando você distribui o aplicativo, você pode escolher que versão a distribuir. Para este exemplo, a versão original é distribuída.

General Settings

* DEPLOYMENT NAME

UpgradeExample

* APPLICATION VERSION

2.0

1.0

2.0

Enter Tag Name

TERMINATE PROTECTION



AGING POLICY

Uma vez que o aplicativo é distribuído pode ser promovido da tela das disposições.

Application Name	State	Environment	Created At	Age	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 Upgrade 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						

O processo de upgrade parte da mais baixa série e acontece um nó de cada vez. Para nosso aplicativo de dois níveis, um servidor da Web Apache é promovido.

Uma vez que isso é terminado, o segundo está promovido. Se você definiu um processo de upgrade para o equilibrador da carga de Nginx, está promovido no dura.