

Aggiorna un'applicazione tramite CloudCenter

Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Definizione processo di aggiornamento](#)

[Crea nuova versione](#)

[Distribuisci applicazione](#)

Introduzione

Questo documento descrive il processo di aggiornamento di un'applicazione tramite CloudCenter.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- CloudCenter
- Bash

Componenti usati

Le informazioni di questo documento si basano su CloudCenter 4.8.1.1.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Esistono diversi modi per aggiornare un'applicazione in CloudCenter. Un'opzione consiste nel creare un'azione personalizzata che può essere applicata a singole VM ed esegue uno script di aggiornamento. Questo metodo offre il controllo completo sull'aggiornamento e consente di testare un nodo prima di aggiornare il nodo successivo. Lo svantaggio è che si tratta di un processo manuale che richiede la scrittura di script personalizzati per ogni aggiornamento. Il metodo preferito consiste nell'utilizzare il framework di aggiornamento di CloudCenter per automatizzare il processo di aggiornamento.

Definizione processo di aggiornamento

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

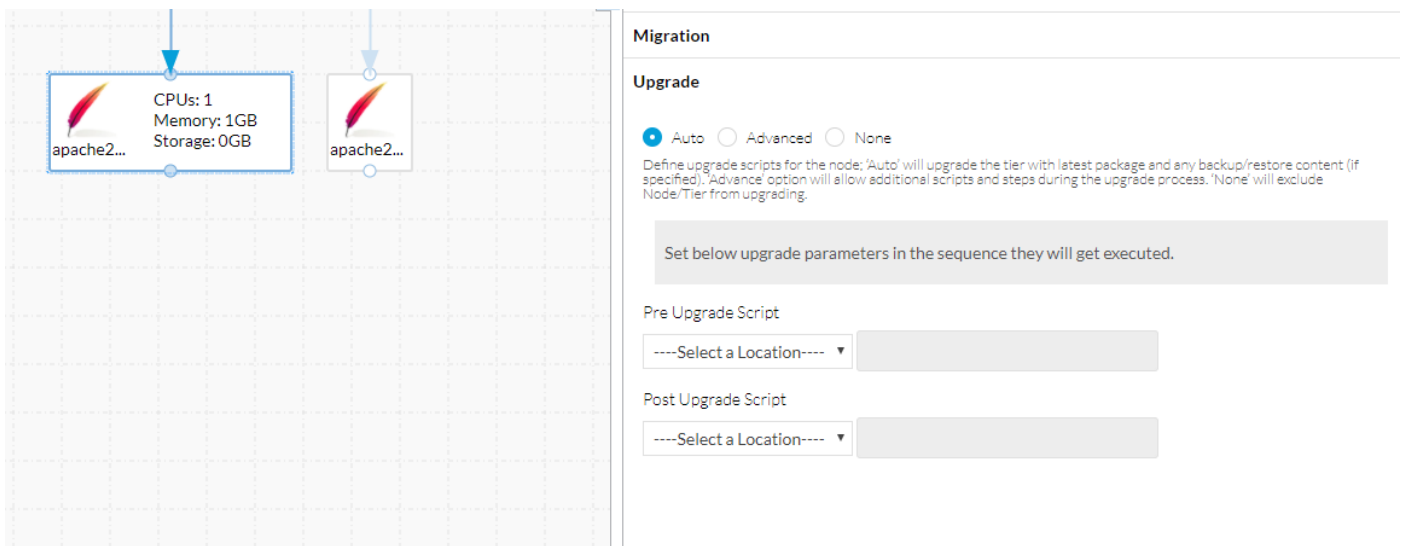
IIS
Web server for Windows-based apps

Jetty
Java-based HTTP server

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

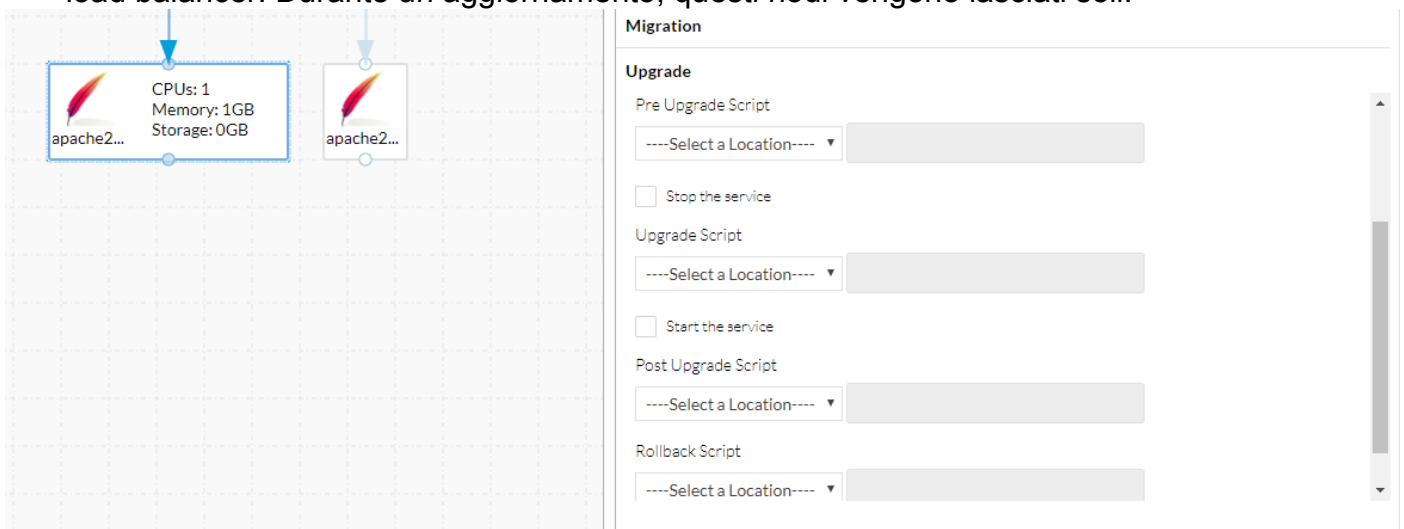
In questa applicazione di esempio sono presenti due server Web Apache dietro un load balancer Nginx. Questi server Web sono identici e forniscono disponibilità HA a un sito Web ospitato. Un processo di aggiornamento ideale consente di aggiornare singolarmente i nodi in modo che vi sia sempre un nodo che ospita il sito Web che consente tempi di attività del 100% durante il processo di aggiornamento.

Per impostazione predefinita, durante un aggiornamento CloudCenter scarica nuovi pacchetti e contenuti, quindi utilizza gli script di backup e ripristino per salvare in modo permanente i dati. Se è necessaria una logica più approfondita, è possibile includere gli script di aggiornamento.



Nella scheda **Migrazione** sono disponibili gli script di backup e ripristino. Questi vengono utilizzati sia per la migrazione che per l'aggiornamento. La scheda **Aggiorna** dispone di tre opzioni: **Automatico**, **Avanzato**, **Nessuno**.

- Auto consente a CloudCenter di aggiornare automaticamente il nodo, scarica il nuovo contenuto ed esegue gli script di backup e ripristino per preservare le informazioni importanti.
- Avanzato consente il controllo completo del processo di aggiornamento.
- Nessuno indica che il nodo non deve essere aggiornato. È possibile eseguire questa operazione per i nodi che non presentano modifiche da una versione all'altra, ad esempio un load balancer. Durante un aggiornamento, questi nodi vengono lasciati soli.



Avanzate consente di aggiungere altri script e di interrompere e avviare il servizio durante l'aggiornamento.

Dopo aver definito tutte le azioni di aggiornamento necessarie, è importante **salvare** l'applicazione prima di passare alla fase successiva

Crea nuova versione

Dopo aver salvato l'applicazione, tornare al **Modellatore topologia**.

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

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

CloudCenter gestisce l'aggiornamento con l'aiuto del controllo delle versioni. L'applicazione nella figura precedente è la versione 1.0, che può essere vista nell'angolo superiore sinistro. Per utilizzare lo strumento di aggiornamento di CloudCenter, è necessario creare una nuova versione.

- Selezionare **Informazioni di base**.
- Immettere una nuova **versione**.

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 salva la versione 1.0 e inserisce tutte le nuove modifiche nella versione 2.0.

Ciò indica a CloudCenter che esiste una nuova versione e consente di tenere traccia delle differenze. Poiché si tratta solo di due server Web, l'unica differenza consiste nell'aggiornare il **pacchetto di applicazioni** in modo che punti a un nuovo file zip.

L'applicazione può essere salvata nuovamente.

Distribuisce applicazione

Quando si distribuisce l'applicazione, è ora possibile scegliere la versione da distribuire. Per questo esempio, viene distribuita la versione originale.

General Settings

* DEPLOYMENT NAME

UpgradeExample

* APPLICATION VERSION

2.0

1.0

2.0

Enter Tag Name

TERMINATE PROTECTION



AGING POLICY

Una volta distribuita, l'applicazione può essere aggiornata dalla schermata Distribuzioni.

Application 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 02:33 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						

Il processo di aggiornamento inizia dal livello più basso ed ha luogo un nodo alla volta. Per l'applicazione a due livelli, viene aggiornato un server Web Apache.

Una volta completato, il secondo viene aggiornato. Se è stato definito un processo di aggiornamento per il servizio di bilanciamento del carico di Nginx, verrà aggiornato nell'ultimo.