Upgrade an Application Using CloudCenter

Contents

Introduction

This document describes the process to upgrade an application using CloudCenter.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- CloudCenter
- Bash

Components Used

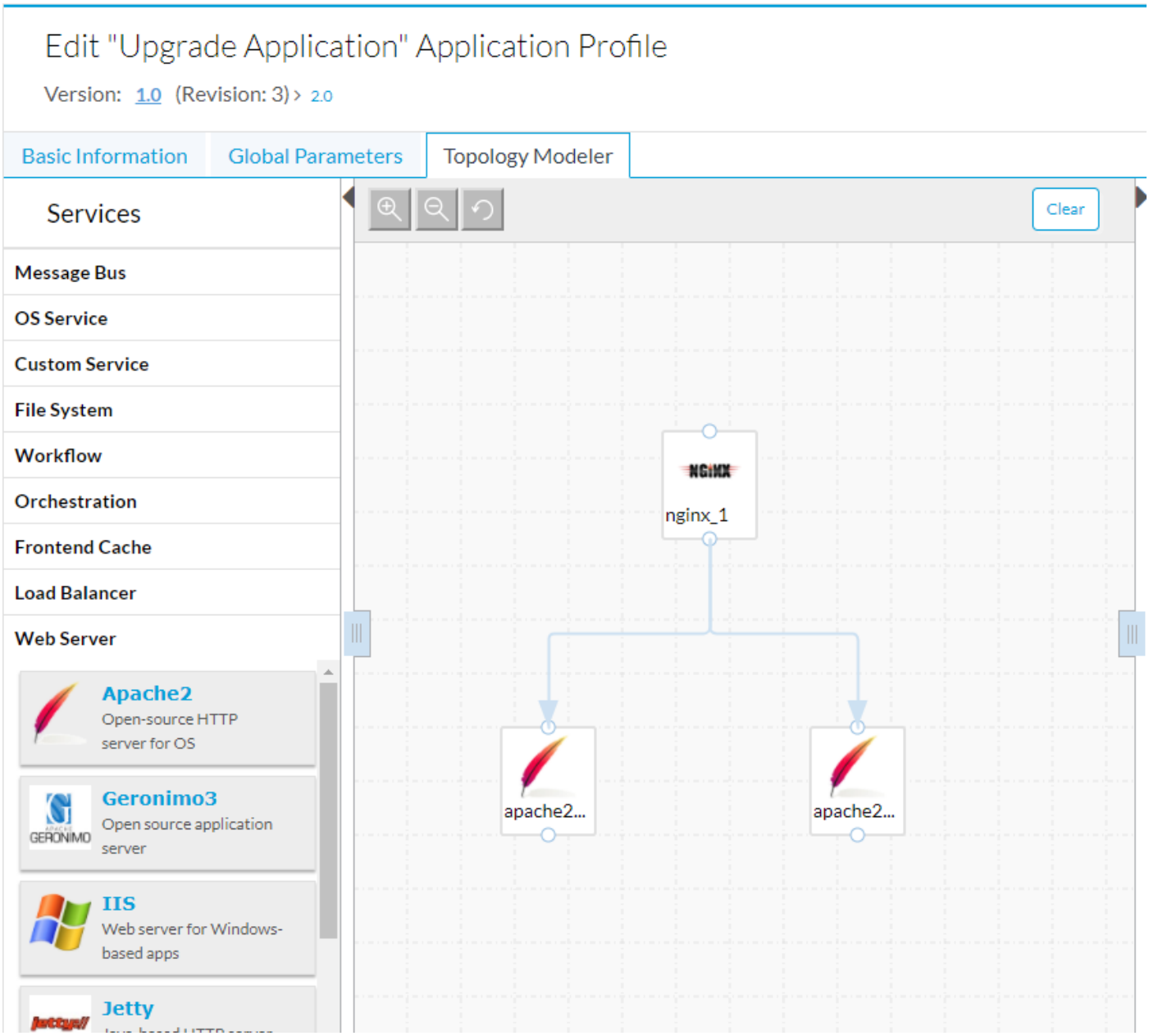
The information in this document is based on CloudCenter 4.8.1.1.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Background Information

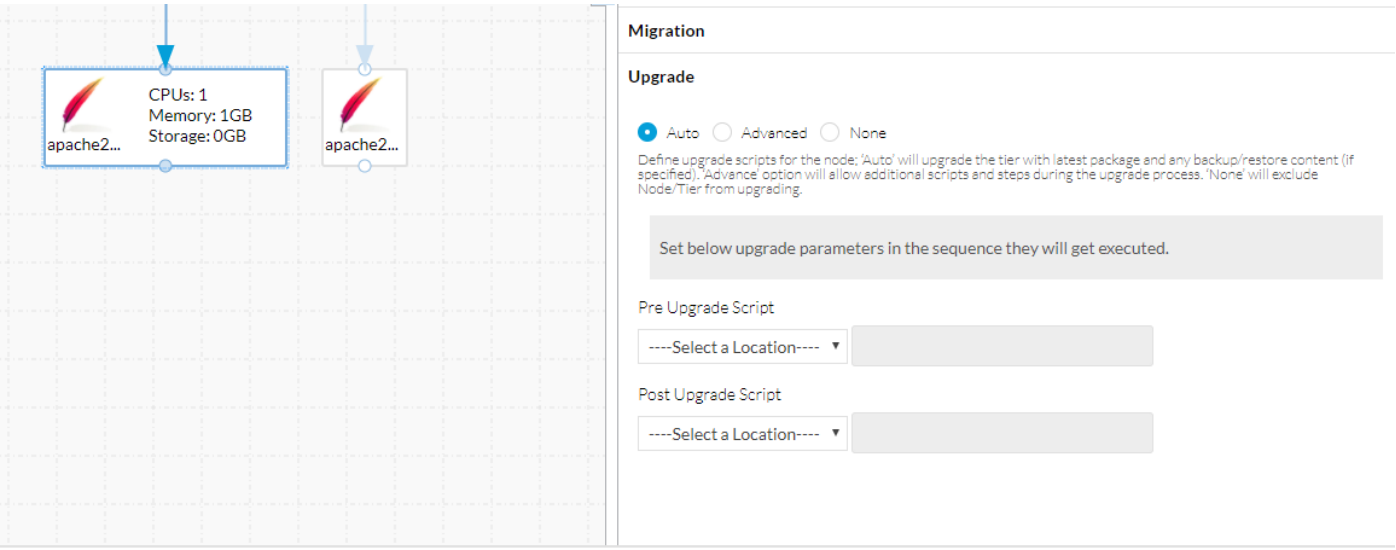
There are multiple ways to upgrade an application in CloudCenter. One option is to create a custom action that can be applied to individual VM's and runs an upgrade script. This method gives you complete control over the upgrade and allows testing of one node before upgrading the next node. The downside is that it is a very manual process that requires writing individualized scripts for every upgrade. The preferred method is to make use of CloudCenter's upgrade framework to automate the upgrade process.

Define Upgrade Process



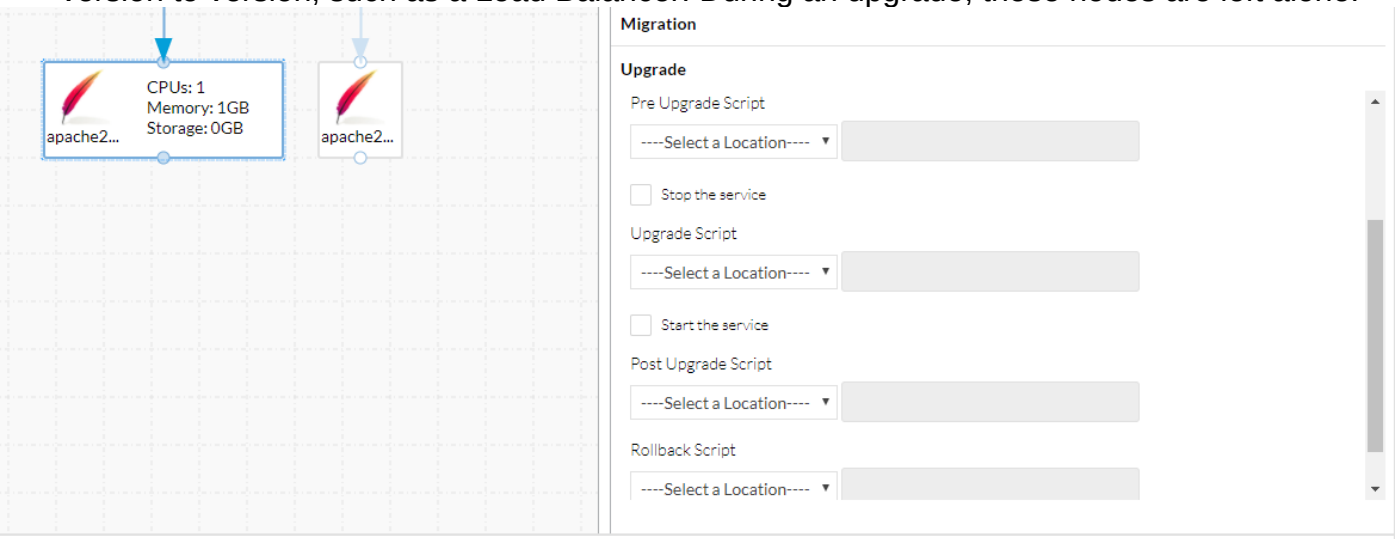
In this sample application, there are two Apache web servers behind a Nginx load balancer. These web servers are identical and provide HA availability to a website that is being hosted. An ideal upgrade process allows the nodes to be upgraded individually so that there is always a node hosting the website allowing for 100% uptime during the upgrade process.

By default, during an upgrade CloudCenter downloads any new packages and content, then make use of any backup and restore scripts to persist data. If more in-depth logic is needed, then upgrade scripts can be included.



Under **Migration** tab, the backup and restore scripts can be found. Those are used both for Migration and Upgrade. The **Upgrade** tab has three options: **Auto, Advanced, None**.

- Auto allows CloudCenter to automatically upgrade the node, it downloads the new content, and runs the backup and restores scripts to preserve important information.
- Advanced allows the complete control of the upgrade process.
- None means do not upgrade this node, it can be done for nodes that have no changes from version to version, such as a Load Balancer. During an upgrade, these nodes are left alone.

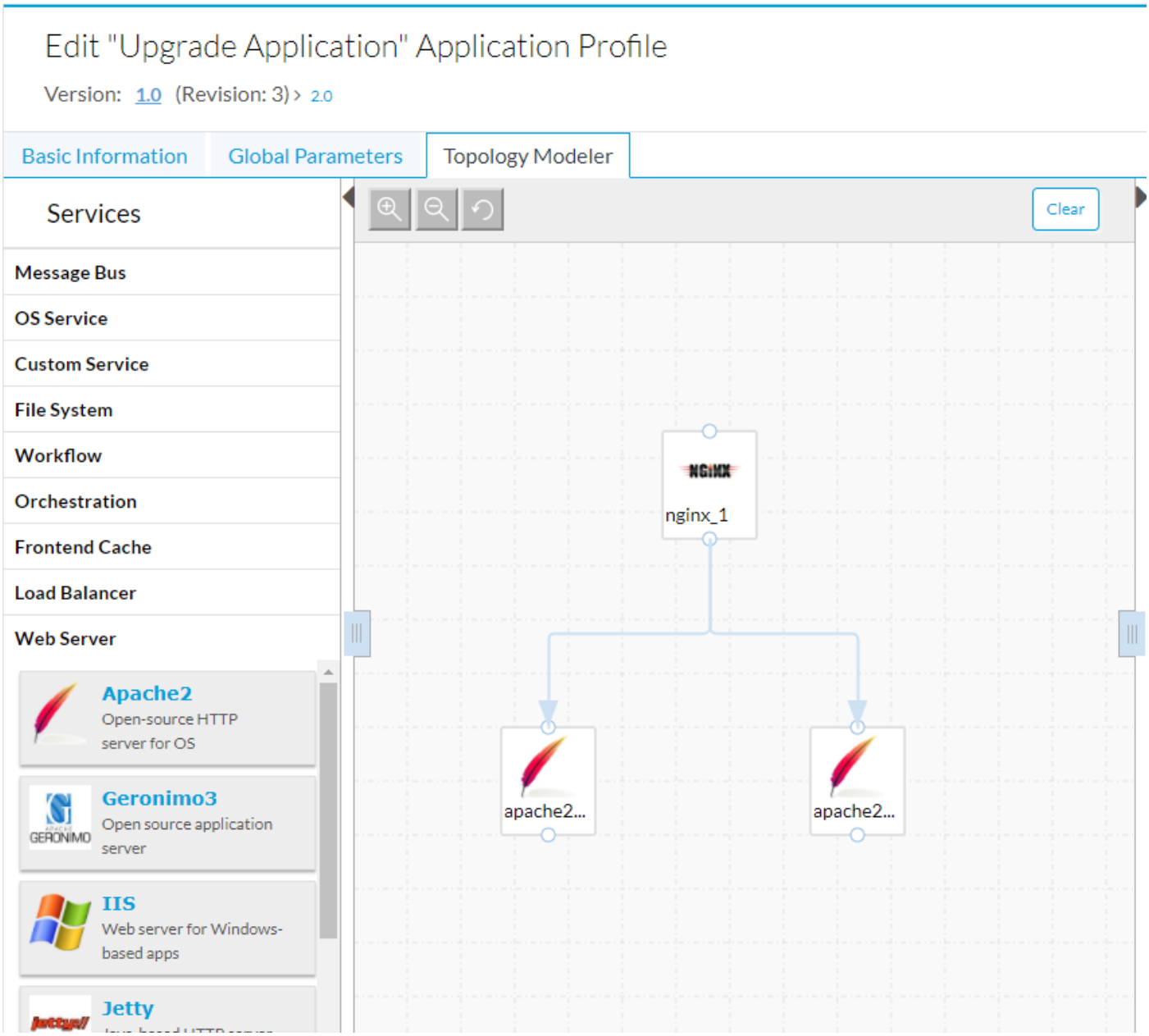


Advanced allows more scripts to be added and allows you to stop and start the service during the upgrade.

Once all necessary upgrade actions are defined, it is important to **save** the Application before moving on to the next step

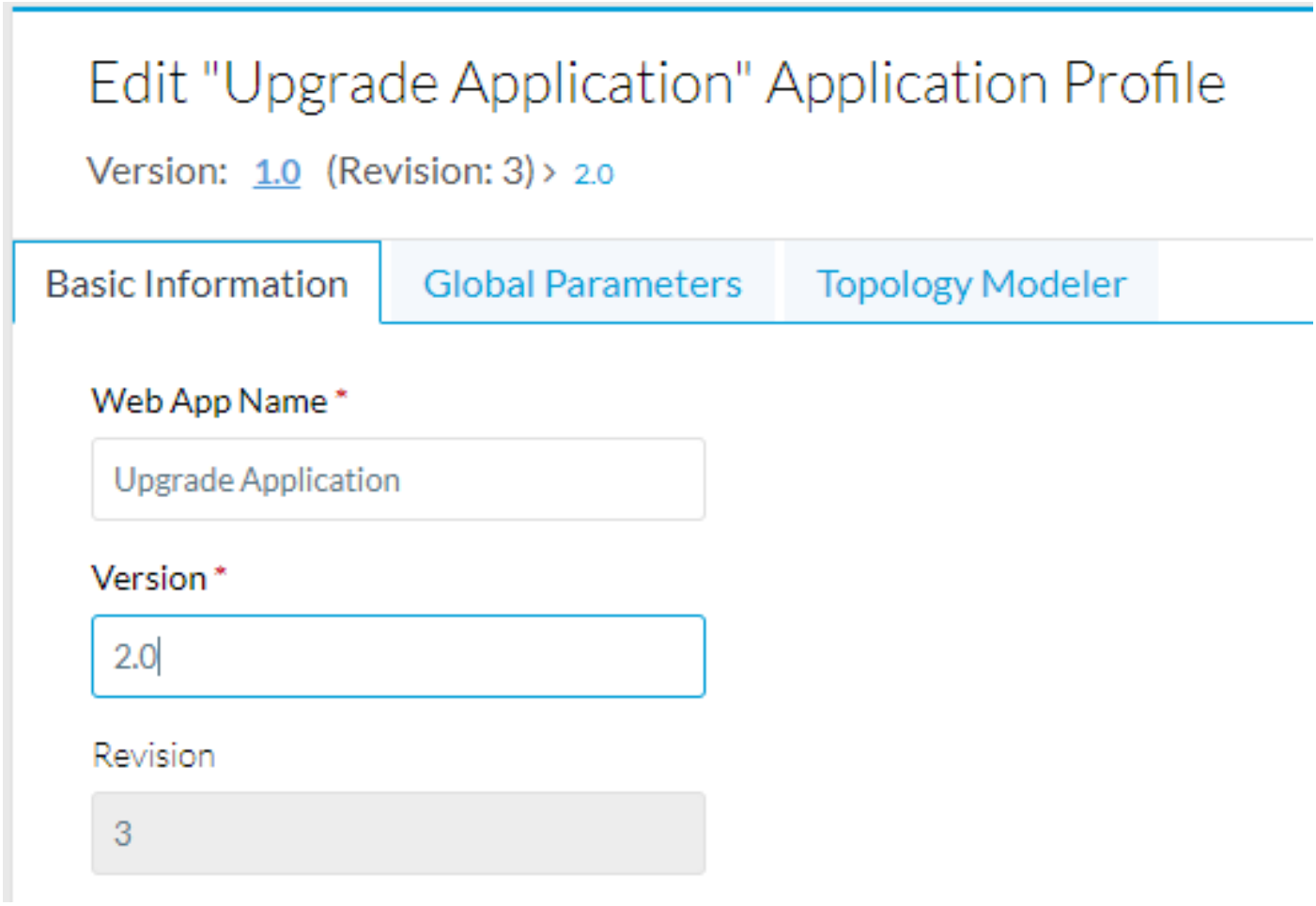
Create New Version

After you save the application, navigate back to the **Topology Modeler**.



CloudCenter handles upgrading with the help of versioning. The application in the picture above is at Version 1.0, this can be seen in the upper left corner. In order to make use of CloudCenter's upgrade tool, a new version must be made.

- Select **Basic Information**.
- Enter a new **Version**.



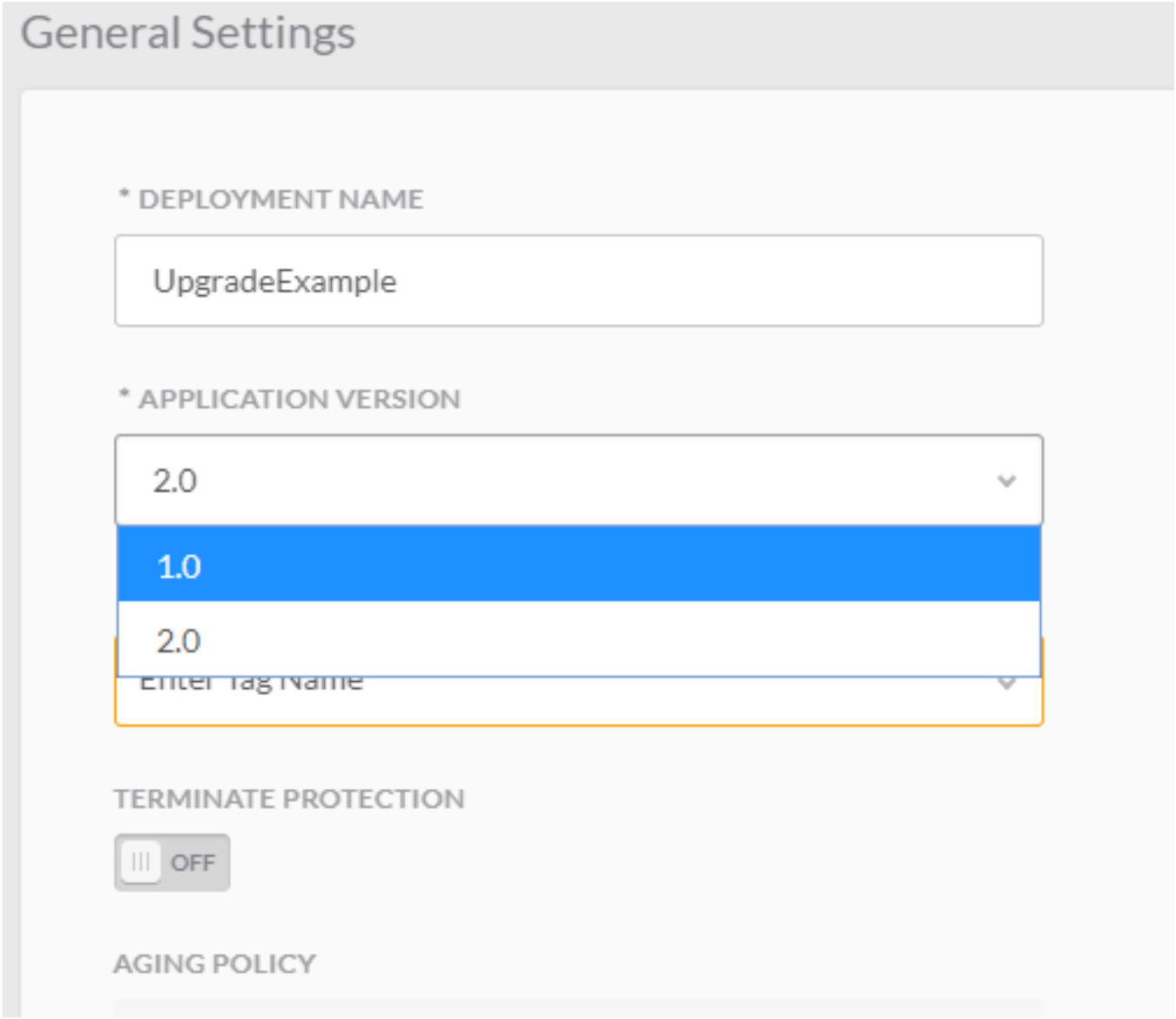
CloudCenter saves Version 1.0 and puts all new changes in Version 2.0.

This tells CloudCenter that there is a new version, and allows it to track the differences. Since this application is just two web servers, the only difference is to update the **Application Package** to point to a new zip file.

The application can be saved again.

Deploy Application

Now, when you deploy the application, you can choose which version to deploy. For this example, the original version is deployed.



Once the application is deployed it can be upgraded from the Deployments Screen.

The upgrade process starts from the lowest tier and happens one node at a time. For our two-tier application, one Apache web server is upgraded.

Once that is completed, the second is upgraded. If you have defined an upgrade process for the Nginx load balancer, it is upgraded in the last.