



Hewlett Packard
Enterprise

Cloud Service Automation

Service Instance Upgrade

Software version: 4.80

Document release date: January 2017

Software release date: January 2017

Contents

Introduction	4
Service instance upgrade process	5
Concepts	7
Upgrade service design.....	7
Upgradable service design.....	7
Upgrade service offering	9
Upgradable service offering	10
Upgradable service subscription	11
Upgrading lifecycle state	11
Lifecycle transition during upgrade of service instances	12
Configure a service design.....	13
Create an upgrade service design.....	13
Create a new service design version.....	13
Establish an upgrade path between two design versions.....	14
Edit an upgrade service design	15
Restrictions on editing upgrade service designs	16
View upgrade paths.....	17
Edit upgrade path description or delete an upgrade path	18
View differences between design versions.....	19
Configure a service offering.....	19
Create an upgrade service offering	20
Create a new service offering version	22
Establish an upgrade path between two offering versions	23
Edit an upgrade service offering	24
View upgrade paths.....	24
Edit upgrade path description or delete an upgrade path	25
Publish an upgrade service offering.....	25
Upgrade a subscription	26
About the service subscription upgrade process	30
Behavior of Dynamic Properties during Service Instance Upgrade	31
Use cases for constructing upgrade service designs	33
Use case 1 – add monitoring capability using HPE SiteScope to server components in service instances	34
Use case 2 – deploy applications using HPE Server Automation to server components in existing service instances	35
Use case 3 – provide additional public actions to service instance components.....	35
Use case 4 – upgrade an application deployed through a sequenced design.....	36
Use case 5 – configure additional firewall rules or apply patches to virtual machines deployed through a sequenced design.	36
Use case 6 – fix problems with modifying or cancelling an existing subscription	36
Failure handling when upgrading a service instance	36
Token to indicate upgrade retry.....	37
REST API to upgrade a service subscription.....	38
Upgrading multiple subscriptions originating from the same service offering	38
Send documentation feedback	40

Legal notices 40

Warranty..... 40

Restricted rights legend..... 40

Copyright notice 40

Trademark notices..... 40

Documentation updates..... 40

Support..... 40

Introduction

HPE Cloud Service Automation (CSA) provides the unique ability to upgrade service instances. CSA service instances can be upgraded in order to provide additional functional capability and additional operational control beyond what is possible based on the current definition of the underlying design. To a *certain extent*, this functionality can also be used to fix any implementation problems present in service instances. In the current version of CSA, the ability to upgrade a service instance is only possible for service instances which are based on sequenced service designs. Service instances can be upgraded to accomplish a variety of use cases. A few examples are discussed below.

Take an example of a sequenced design that is capable of deploying one or more virtual machines based on the VMWare vCenter resource provider. In the service design, virtual machine settings (such as number of CPUs, amount of memory and disk size) are configured as properties on the server component. This service design can also have subscriber options related to compute size (such as small, medium or large) which can be used for ordering virtual machines and for modifying service subscriptions so that virtual machines can be of a different compute size. Subscriber option model properties (such as number of CPU, amount of memory and disk size) are bound to the respective server properties. Once subscriptions are created based on this service design, possible operations only include modification of subscription by choosing a different option and executing public actions on server components to start/stop/restart/suspend virtual machines.

After creation of subscriptions based on this service design, there might be additional needs such as adding capability to monitor virtual machines deployed through subscriptions, deploying additional firewall rules on top of existing firewall rules on virtual machines, deploying applications on top of the virtual machines deployed through the service subscriptions, or introducing additional public actions on server components in order to provide more operational control. Examples of these actions could be an action to take snapshots of virtual machines or creating ovf templates out of the virtual machines. All such needs are beyond the definition of the original service design underlying the service instances. Needs such as the ones discussed above can be accomplished through the service instance upgrade functionality available in CSA.

Take another example of a business setting where a sequenced service design is being utilized for deploying multi-tiered applications. Newer business requirements would result in the need to move to newer versions of databases and newer versions of applications. They would also result in change of application deployment architecture. Service instance upgrade functionality will also enable upgrading service instances representing multi-tiered applications to account for changes in database versions, application versions and application deployment architecture.

CSA ships with sample designs which enable upgrading service instances for some of the use cases discussed above. A different section of this white paper discusses implementation aspects of upgrading service instances using these sample designs.

Note: Throughout this document, we will be talking about upgrading service instances and service subscriptions interchangeably. Both of them are one and the same.

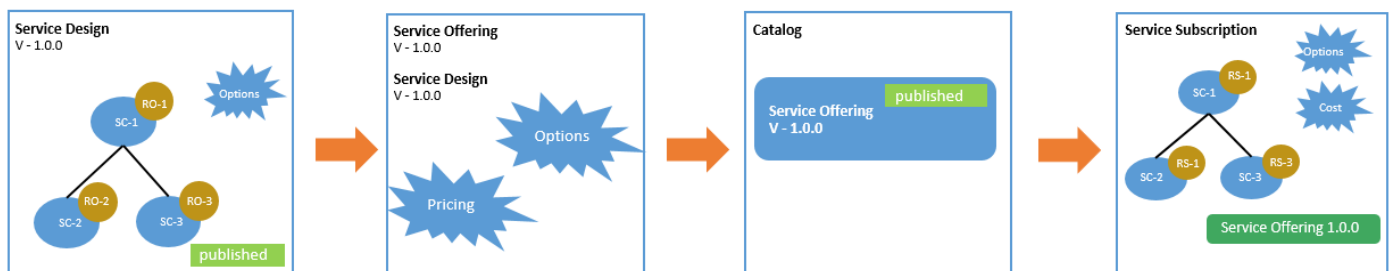
Service instance upgrade process

This section talks about the overall process involved in upgrading service instances: creating a service subscription and upgrading a service subscription. Each of the individual steps involved in *upgrading* service subscriptions are covered at a greater depth later in this document.

The steps to *create* service subscriptions in CSA are as follows:

1. The Service Designer creates a service design that can orchestrate the deployment of a cloud service. All service design components and resource offerings have a unique identifier invisible to end users. When a service design is published, it cannot be edited and service offerings can be created from it.
2. The Service Business Manager creates a service offering from the service design. A service offering can be associated with a price, can have customized subscriber options, and can be associated with SLA documents.
3. The Service Business Manager publishes the service offering to a catalog. Approval policies may be added to the service offering in the catalog.
4. A subscriber orders a service offering from the Marketplace Portal which results in the creation of a service subscription. During the service subscription creation process, service instance components and resource subscriptions (the run-time representation of resource offering) inherit the unique identifier from the service design's components and resource offerings respectively (that is, service component and resource offering identifiers match between the service design and service instance). A service instance associated with the service subscription represents the deployed topology of a cloud service.

Figure 1. Service subscription creation



Once service subscriptions have been created, the steps to upgrade service subscriptions are as follows:

1. After analyzing additional capabilities and operational control that needs to be provided to service instances on top of what is available, the Service Designer creates a new version of a service design *cloned* from the original version of the service design. The cloning process will ensure that the service component's unique identifiers are copied over to the service components of the new version of the service design. These unique identifiers will play a major role in the service instance upgrade process.

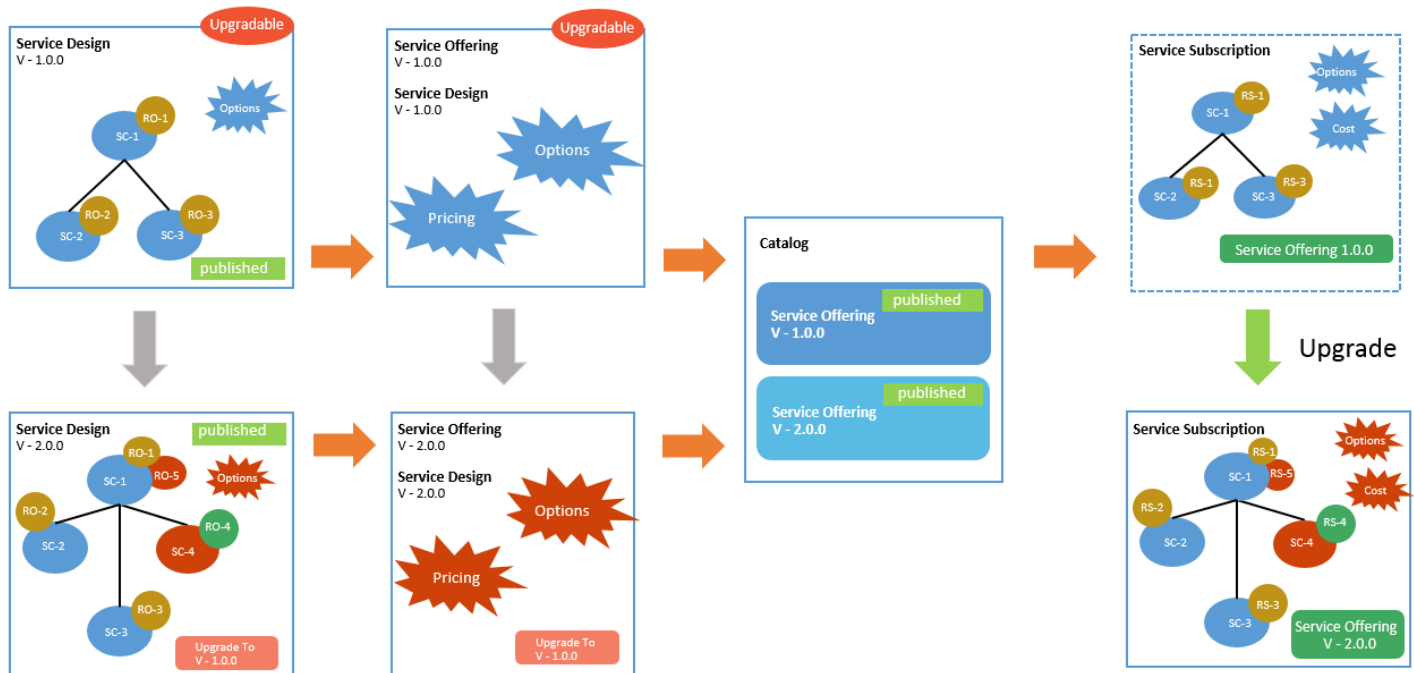
The new version of the service design is called an upgrade service design while the original version of the service design is called an upgradable service design. The changes necessary for providing additional capability or operational control are added to the new version of the service design. While the upgrade service design can be modified in a variety of ways based on the use case of service instance upgrade, not all changes are permitted. Changes that can be made for different use cases and changes that are not allowed on upgrade versions will be discussed in a later section of this document. When an upgrade service design is published, it cannot be edited and service offerings can be created from this service design.

2. The Service Business Manager creates a new version of a service offering *cloned* from the original version of the service offering based on the upgrade version of the service design. The new version of the service offering is called an upgrade service offering while the original version of the service offering is called an upgradable offering. Subscriber options, pricing and SLA documents can be customized in the new version of the service offering.
3. The upgrade service offering is published into the same catalog as the upgradable the service offering.

Note: This step is required. If the upgrade service offering is not published, the service subscription cannot be upgraded.

4. The Operations area of the Cloud Service Management Console indicates that service subscriptions originating from the original version of service offering are upgradable. The Service Operations Manager can upgrade service subscriptions based on the new version of the service offering from the Operations area or through REST APIs. During the service instance upgrade process, the unique identifiers from service instance components will be utilized for detecting changes that need to be merged from the upgradable design.

Figure 2. Service instance upgrade process



Concepts

Upgrade service design

An upgrade service design is a new version (clone) of a service design created from and *based on* an existing version of a service design for the purpose of creating new service subscriptions or upgrading service subscriptions originating from the existing version of the service design.

In the Sequenced Design Version Overview area of the Cloud Service Management Console, upgrade service designs have an **Upgradable From** field populated with the version of the existing service design that it is based on, indicating that service instances originating from that existing service design version can be upgraded to this new version.

Figure 3. The **Upgradable From** field of service design version 2.0.0 has been populated with version 1.0.0 indicating version 2.0.0 is going to act as an upgrade design for service subscriptions originating from service design version 1.0.0

CLOUD SERVICE AUTOMATION admin Help

Sequenced Designs

Simple Compute Linux (2.0.0)

Overview Designer Subscriber Options Upgradability

This design is part of an upgrade path and has restricted editability and actions.

Simple Compute Linux

Description
Simple Compute Linux

Version
2.0.0

URL

Published
No

Upgradable From
1.0.0

Upgradable To
No upgrade path defined

Service Composite
Cloud
Infrastructure Service
Cloud
Server Group
Cloud
Server

Upgradable service design

An upgradable service design is an existing version of a service design where service subscriptions originating from the existing design can be upgraded to a different version of the existing service design. A service design is upgradable only in the presence of an upgrade service design (a new version of the existing service design).

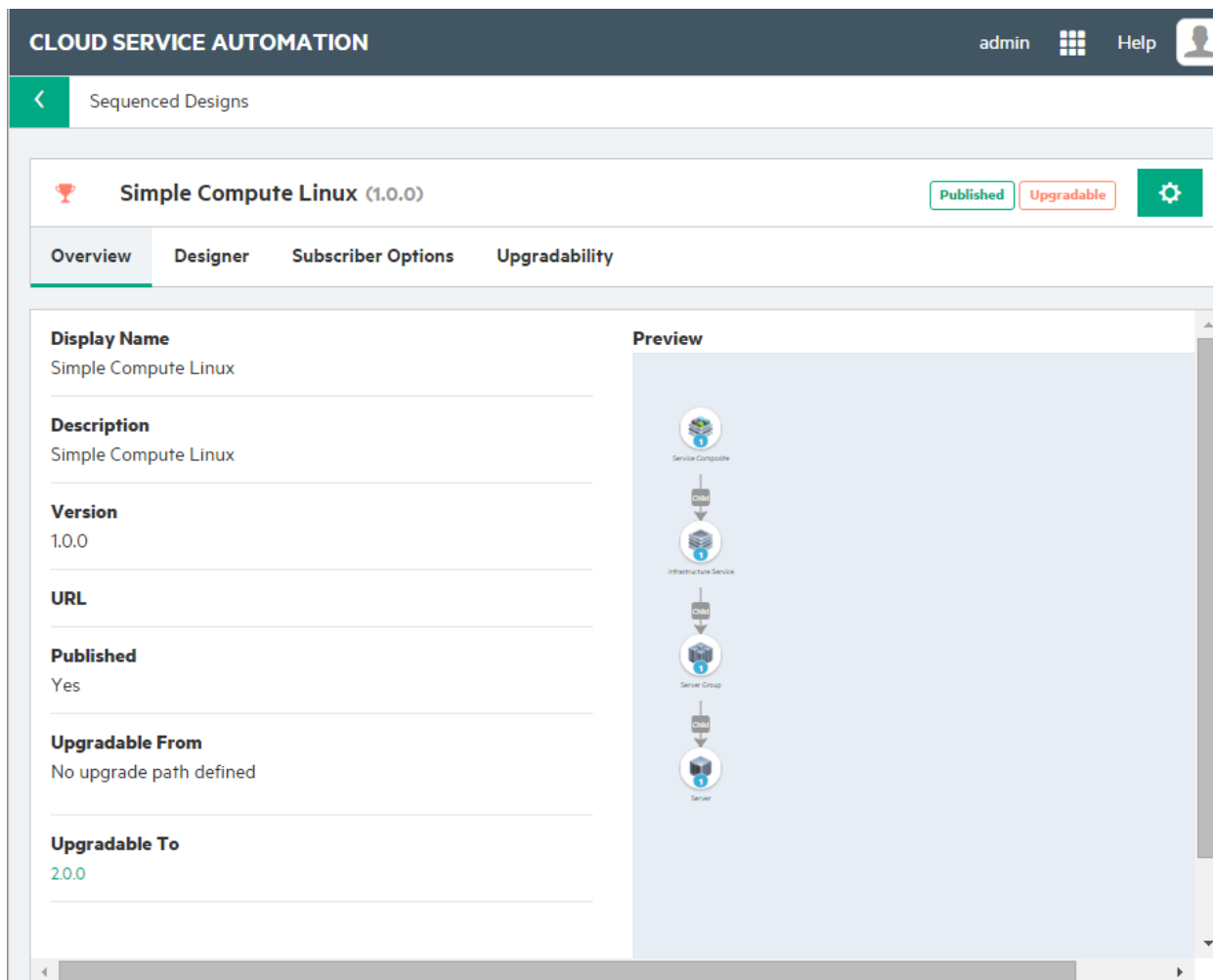
In the service design page of the Cloud Service Management Console, upgradable service designs have an **Upgradable** label in the right corner.

Figure 4. The **Upgradable** label in the service design page



In the service design version page of the Cloud Service Management Console, upgradable service designs have an **Upgradable To** field populated with the versions of the service design which can be used for upgrading service subscriptions originating from it. Upgradable service designs also have an **Upgradable** label in the top right corner.

Figure 5. The **Upgradable To** field of service design version 1.0.0 has been populated with version 2.0.0 indicating service subscriptions originating from version 1.0.0 are upgradable to version 2.0.0; An **Upgradable** label is also displayed in the top right corner



Upgrade service offering

An upgrade service offering is a new version of a service offering created from an existing version of service offering that is based on an upgrade service design. The upgrade service offering is used to create new subscriptions or upgrade service subscriptions originating from the existing version of the service offering.

In the **Overview** tab of the Offerings area of Cloud Service Management Console, upgrade service offerings have **Upgradable From** field populated with the version of the service offering that they are based on and are going to be used for upgrading service subscriptions originating from that service offering version.

Figure 6. The **Upgradable From** field of service offering version 2.0.0 has been populated with version 1.0.0 indicating version 2.0.0 is going to act as an upgrade offering for service subscriptions originating from service offering version 1.0.0

The screenshot displays the 'Simple Compute Linux (2.0.0)' service offering page in the Cloud Service Automation console. The page has a dark blue header with 'CLOUD SERVICE AUTOMATION', 'admin', 'Help', and a user profile icon. A green sidebar on the left shows 'Offerings'. The main content area has a tabbed interface with 'Overview', 'Publishing', 'Options', 'Pricing', 'Documents', 'Screenshots', 'Versions', and 'Upgradability'. The 'Overview' tab is active, showing a list of fields: 'Version Name' (2.0.0), 'Description' (Simple Compute Linux - 2.0.0), 'Selected Service Design' (Simple Compute Linux, 2.0.0), 'Upgradable From' (1.0.0), and 'Upgradable To' (None). A large red trophy icon is displayed on the right side of the page, and a message states 'No tags associated with this offering'.

Field	Value
Version Name	2.0.0
Description	Simple Compute Linux - 2.0.0
Selected Service Design	Simple Compute Linux 2.0.0
Upgradable From	1.0.0
Upgradable To	None

Upgradable service offering

An upgradable service offering is a version of a service offering where service subscriptions originating from it have to be upgraded to a different version of the service offering. A service offering version is upgradable only in the presence of upgrade service offerings.

In the **Overview** tab of the Offerings area of Cloud Service Management Console, upgradable service offerings have an **Upgradable To** field populated with the versions of service offerings which are going to be used for upgrading service subscriptions originating from the current version of the service offering. Upgradable service offerings also have an **Upgradable** label in the top right corner. Upgradable service offerings also have an **Update Available** label indicating the presence of upgrade service designs on which upgrade service offerings are based.

Note: Service subscriptions originating from an upgradable service offering can be upgraded to many version of upgrade service offerings if there are multiple versions of upgrade service offerings.

Figure 7. The **Upgradable To** field of service offering version 1.0.0 has been populated with version 2.0.0 indicating service subscriptions originating from version 1.0.0 are upgradable to version 2.0.0; An **Upgradable** label is also populated in the top right corner

CLOUD SERVICE AUTOMATION admin Help

< Offerings

Simple Compute Linux (1.0.0) Upgradable

Overview Publishing Options Pricing Documents Screenshots Versions Upgradability

Simple Compute Linux

Version Name
1.0.0

Description
Simple Compute Linux - 1.0.0

Selected Service Design
Simple Compute Linux
1.0.0

Upgradable From
None

Upgradable To
2.0.0

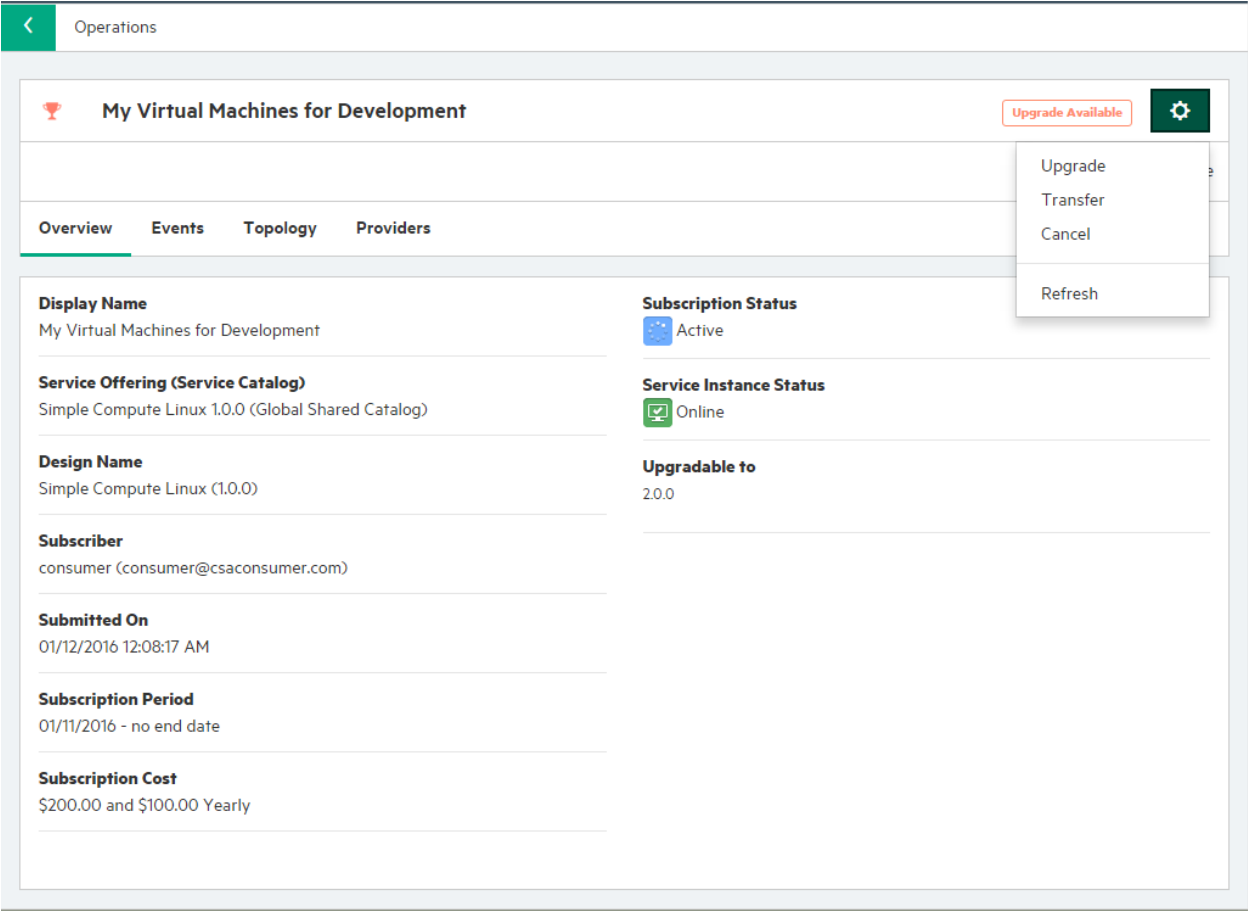
Tags
No tags associated with this offering

Upgradable service subscription

A service subscription is upgradable if the service subscription is in an Active state and there are upgrade service offerings published into the same catalog as the service offering from which service subscription originated.

In the subscription overview in the Operations area of Cloud Service Management Console, an **Upgrade Available** label indicates that the service subscription is upgradable. The subscription overview indicates that upgrades are available through the **Upgradable to** field. Click the gear icon and select **Upgrade** to upgrade the service subscription.

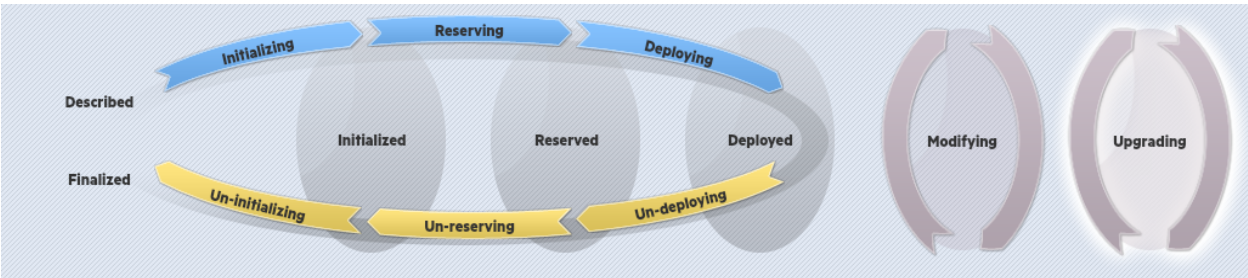
Figure 8. An upgradable service subscription



Upgrading lifecycle state

A new *Upgrading* lifecycle state is used to facilitate the service instance upgrade process. The Upgrading lifecycle state can be utilized for adding lifecycle actions to service components and resource offerings common to the upgrade and upgradable service designs. These lifecycle actions make changes to an existing service component or resource offering in a service instance during the upgrade process.

Figure 9. Upgrading lifecycle state



Lifecycle transition during upgrade of service instances

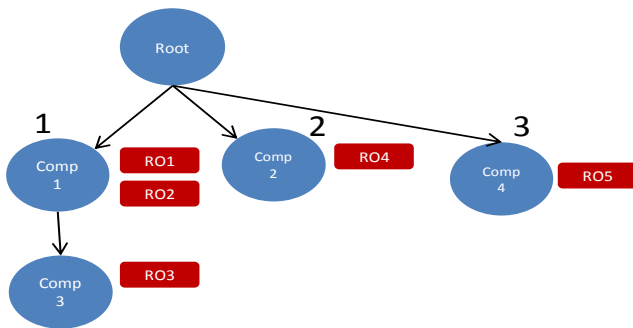
Just like ordering a subscription or modifying a subscription, the service instance undergoes a lifecycle transition by executing various actions configured on components and resource offerings during the process of upgrading subscriptions. Service instance lifecycle transition rules during an upgrade are different from those of ordering or modifying a subscription. They can be summarized as follows:

- Actions configured in the Upgrading lifecycle state are executed for:
 - existing service components in the service instance and
 - existing and new resource offerings added to the existing service component.
- Actions configured in the Initializing, Reserving and Deploying lifecycle states are executed for service components (and resource offerings bound to service components) which are new to the service instance.

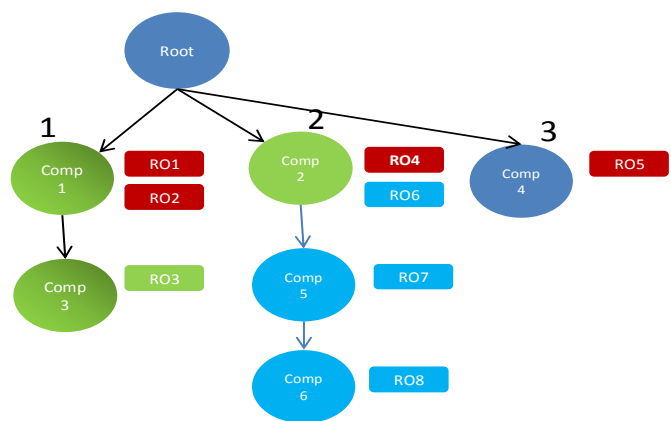
The following figures depict changes made between upgradable and upgrade service designs followed by the lifecycle transition in the service instance during the upgrade process.

Figure 10. Service design version 2.0.0 is an upgrade for service design version 1.0.0

Service Design Version 1.0.0



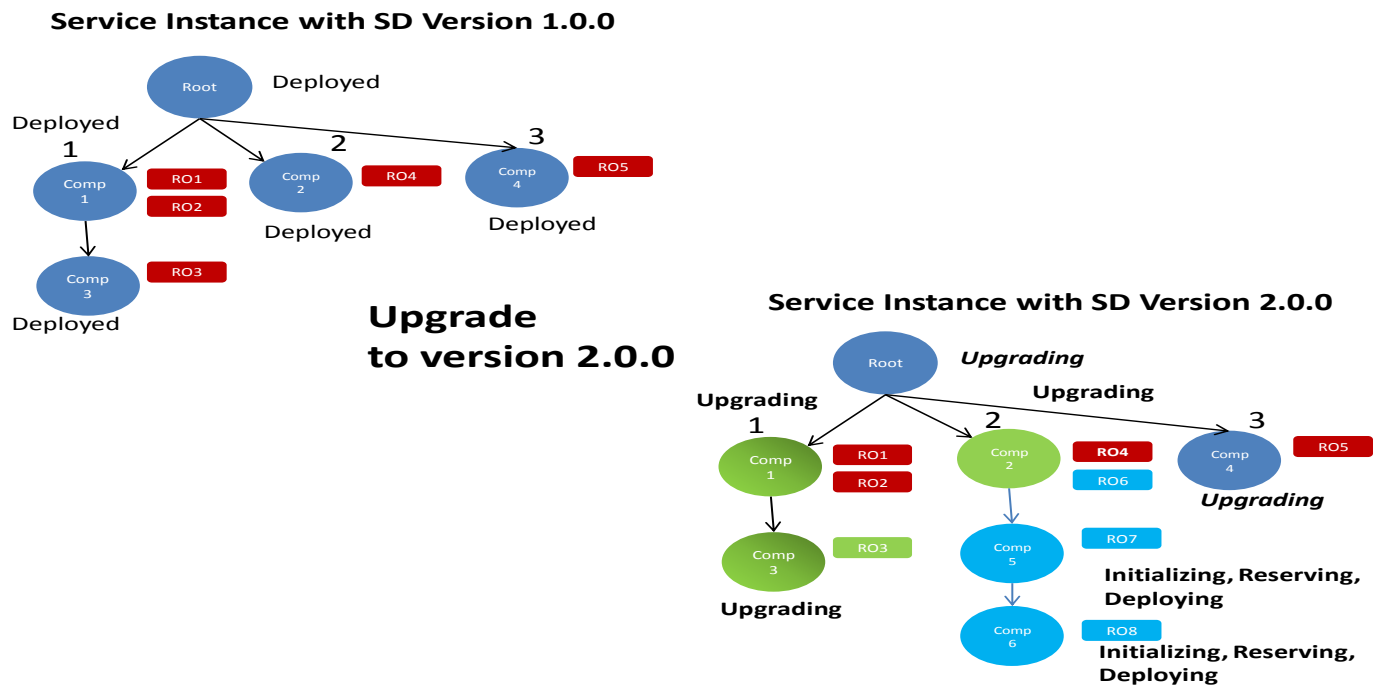
Service Design Version 2.0.0



Changes in Service Design Version 2.0.0

- New property has been added to Comp 1
- New property has been added to Comp 3
- RO 3 in Comp 3 has a new OO flow in Upgrading lifecycle state
- New RO 6 has been added to Comp 2
- Comp 5 and Comp 6 are new components

Figure 11. During an upgrade to design version 2.0.0, actions present in the Upgrading lifecycle state are executed for existing service components (and existing or new resource offerings bound to them) while actions present in the Initializing, Reserving and Deploying lifecycle states are executed for only new service components (and resource offerings bound to them).



Configure a service design

The first step to upgrade a service instance is to configure an upgrade service design:

1. [Create an upgrade service design.](#)
2. [Edit an upgrade service design.](#)

Additional tasks for service design upgrades include the following:

- [View upgrade paths.](#)
- [Delete an upgrade path.](#)
- [View differences between design versions.](#)

Create an upgrade service design

There are two ways to create an upgrade service design for upgrading service instances:

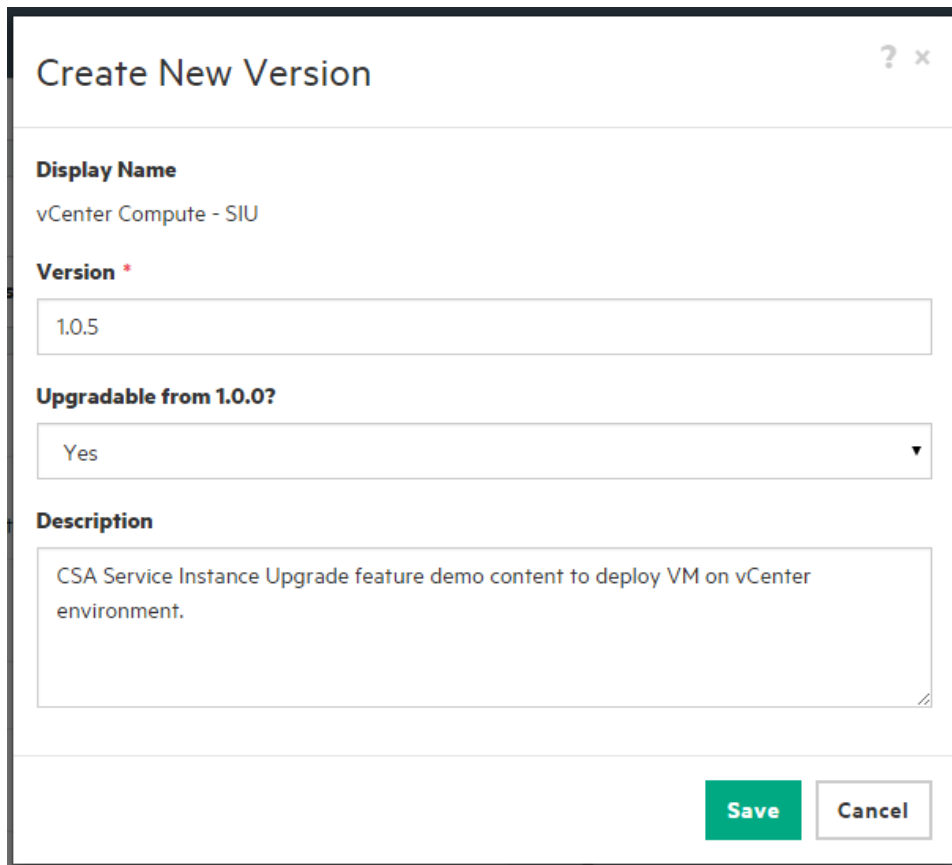
- [Create a new service design version from an existing published design version.](#)
- [Establish an upgrade relationship between two design versions using the Upgradability tab.](#)

Create a new service design version

To create a new service design version from an existing published version:

1. Open the existing design, the design for which you wish to create an upgrade design. This design must be in a published state.
2. Click the gear icon and select **Create New Version**.

3. Enter a new version (required) and select **Yes** for **Upgradable from <base design version>?**.



The 'Create New Version' dialog box is shown. It has a title bar with a question mark and a close button. The form contains the following fields:

- Display Name:** vCenter Compute - SIU
- Version *:** 1.0.5
- Upgradable from 1.0.0?:** Yes (selected in a dropdown menu)
- Description:** CSA Service Instance Upgrade feature demo content to deploy VM on vCenter environment.

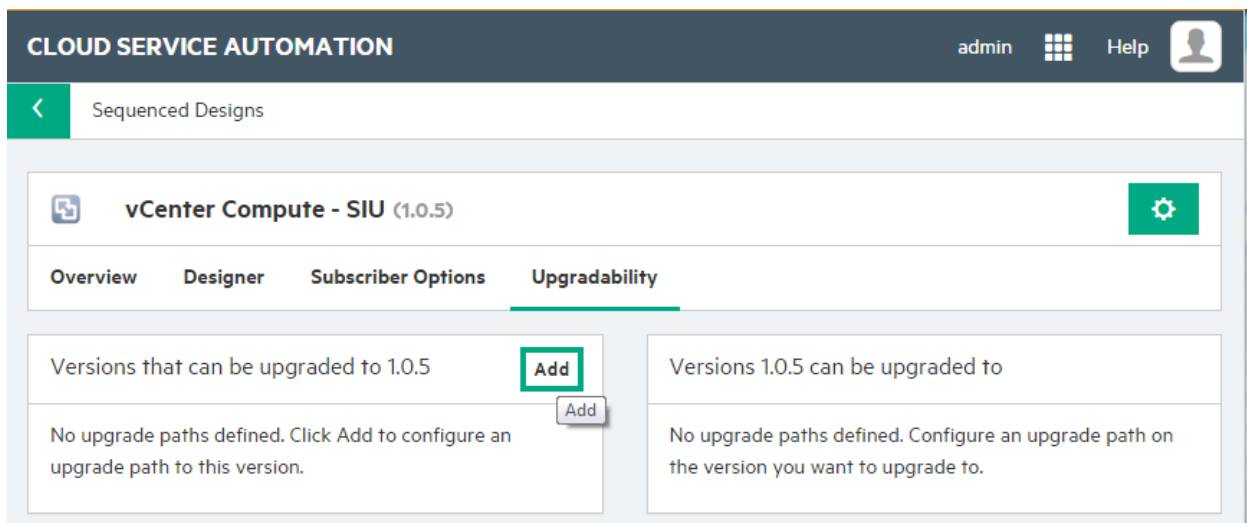
At the bottom right, there are two buttons: **Save** (green) and **Cancel** (white).

4. Click **Save**. This creates an upgrade design and establishes an upgrade path or relationship between the two designs.

Establish an upgrade path between two design versions

The second way to create an upgrade service design is to establish an upgrade path or relationship between two versions of a service design:

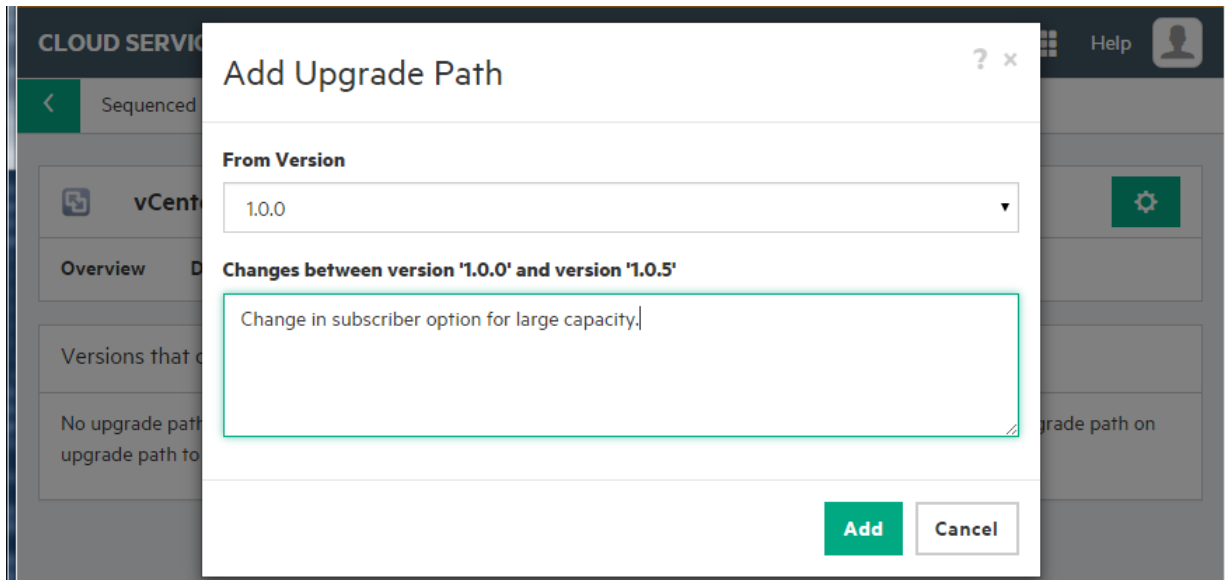
1. Open the design that is to be the upgrade service design. The selected version of the service design *must* be cloned from the existing design from which the upgrade path will be established.
2. Select the **Upgradability** tab and click **Add**. This is only possible if an upgrade is not available from the selected version of the service design to a different version of the service design.



The screenshot shows the 'Upgradability' tab in the Cloud Service Automation interface. The header bar includes 'CLOUD SERVICE AUTOMATION', 'admin', 'Help', and a user icon. The breadcrumb is 'Sequenced Designs'. The main content area is for 'vCenter Compute - SIU (1.0.5)' and has tabs for 'Overview', 'Designer', 'Subscriber Options', and 'Upgradability' (which is selected). Below the tabs, there are two panels:

- Left Panel:** 'Versions that can be upgraded to 1.0.5'. It contains an 'Add' button and a message: 'No upgrade paths defined. Click Add to configure an upgrade path to this version.'
- Right Panel:** 'Versions 1.0.5 can be upgraded to'. It contains an 'Add' button and a message: 'No upgrade paths defined. Configure an upgrade path on the version you want to upgrade to.'

3. Select a design version and add a description of the changes between the two designs. Only design versions which are in a published state will be selectable in the dialog.

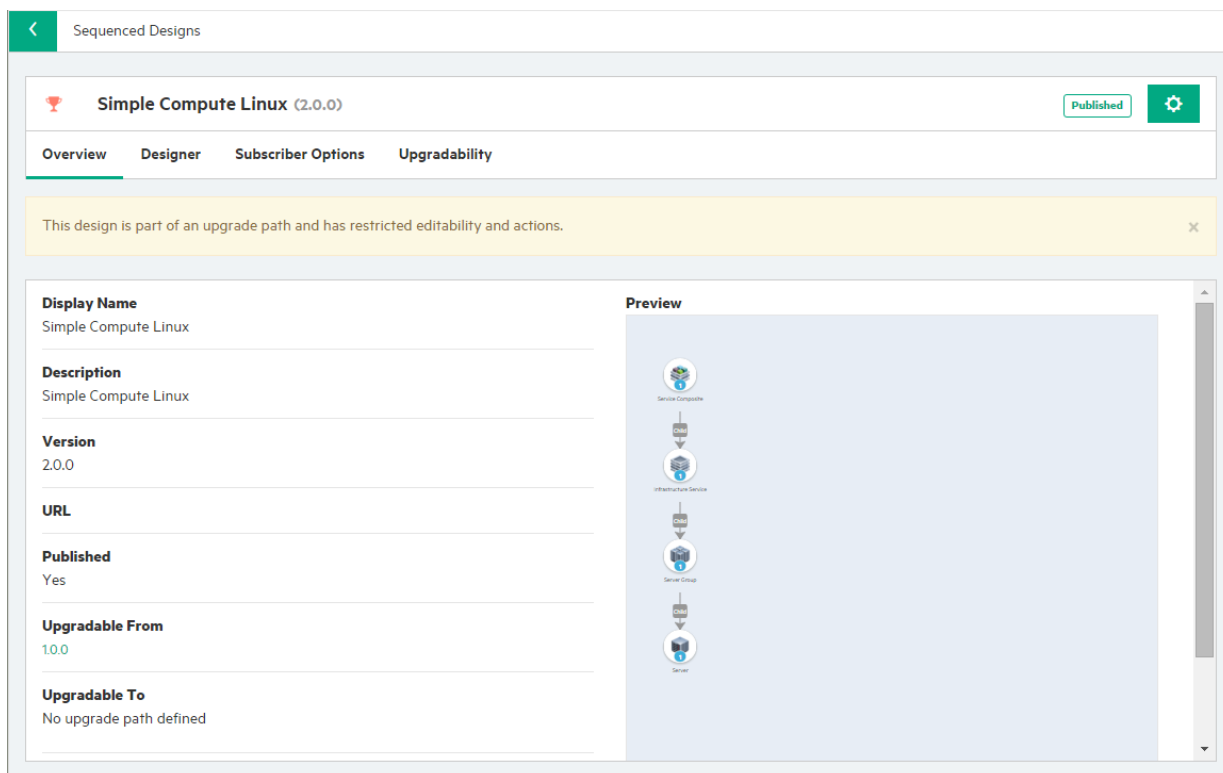


4. Click **Add**. This establishes an upgrade relationship between the two designs. If the current version of a service design violates upgradability rules, an upgrade relation will not be established between the designs. An error message is displayed and additional details of the error are found in the CSA logs.

Edit an upgrade service design

After the upgrade service design is created, changes can be made to it and its subscriber options. However, an upgrade service design has restricted editability and actions, and a message with this restriction is displayed in its service design version page. The sequenced designer and subscriber options interfaces automatically restrict changes that are not allowed. Changes which are not allowed on upgrade service designs are described in the next section. The upgrade service design must be published before an upgrade service offering can be created.

Figure 12. Upgrade version of service design has editability restrictions



Restrictions on editing upgrade service designs

The following is a list of items and descriptions of functions that are restricted:

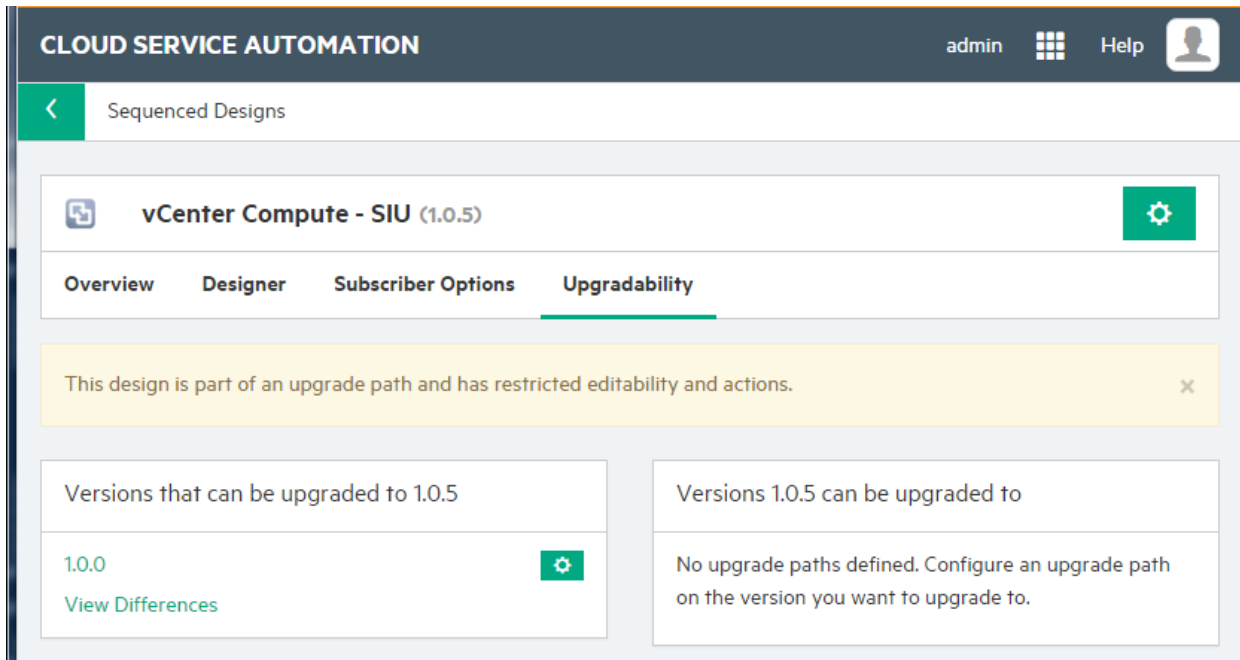
- Resource Offering Properties
 - *Deleting resource offering properties* - The properties of a resource offering cannot be deleted if the resource offering is linked to a service component that is part of an upgrade service design. That is, if a resource offering has been configured for a resource binding of a service component that is part of an upgrade service design, the properties of the resource offering cannot be deleted.
- Service Components and Properties
 - *Deleting a service component* - A service component cannot be deleted if it is part of an upgrade design and the component originated in the upgradable design. A service component can be deleted from the current design if the component originated in the current design (the component was initially added to the current design).
 - *Editing service component attributes: Component Order and Pattern* - The Component Order and Pattern attributes of a service component cannot be edited if the component originated in an upgradable design. The Component Order and Pattern attributes of the component are editable if the component originated in the current design.
 - *Deleting a service component property* - A service component property cannot be deleted if it is part of an upgrade design and the property originated in the upgradable design. A service component property can be deleted from the current design if the service component property originated in the current design (the custom service component property was initially added to the current design).
 - *Editing a service component property: Name* - The Name attribute of a component property cannot be edited if the component is part of an upgrade design and originated in an upgradable design. The Name attribute of a component property can be edited if the component property originated in the current design.
- Service Component Resource Bindings and Properties
 - *Deleting a resource binding* - A resource binding in a service component cannot be deleted if it is part of an upgrade design and the resource binding originated in the upgradable design. A resource binding can be deleted from a component in a design if the resource binding originated in a component of the current design (the resource binding was initially added to a component of the current design).
 - *Editing a resource binding attribute: Binding Order* - The Binding Order of a service component resource binding cannot be edited if the resource binding is part of an upgrade design and originated in an upgradable design. The Binding Order attribute of a resource binding can be edited if the resource binding originated in a component of the current design.
 - *Adding or deleting measurable properties* - A measurable property of a resource binding in a service component cannot be added or deleted if it is part of an upgrade design.
- Subscriber Option Sets
 - *Deleting a subscriber option set* - Subscriber option sets cannot be deleted if the design is part of an upgrade path and the option sets were added to the design to which this design is an upgrade for. If an option set has been added to the current design (whether or not the design is part of an upgrade path), those option sets can be deleted.
- Subscriber Options and Properties
 - *Deleting a subscriber option* - Subscriber options cannot be deleted if the design is part of an upgrade design and the options originated from upgradable design. If an option has been added to the current design (whether or not the design is part of an upgrade path), those options can be deleted.
 - *Deleting a subscriber option property* - Subscriber option properties cannot be deleted if the design is part of an upgrade design and option properties originated from upgradable design. If an option property has been added to the current design (whether or not the design is part of an upgrade path), those option properties can be deleted.
 - *Editing a subscriber option property: name* - The subscriber option property name cannot be edited if the design is part of an upgrade design and option properties originated from upgradable design. If an option property has been added to the current design (whether or not the design is part of an upgrade path), the option property name can be edited.

- *Editing a subscriber option property: target binding* - Target bindings cannot be edited if the design is part of an upgrade design and bindings originated from upgradable design. If a target binding has been added to the current design (whether or not the design is part of an upgrade path), those bindings can be deleted.

View upgrade paths

The **Upgradability** tab displays the versions of service designs that can be upgraded to the current version as well as the versions of service designs to which the current version can upgrade.

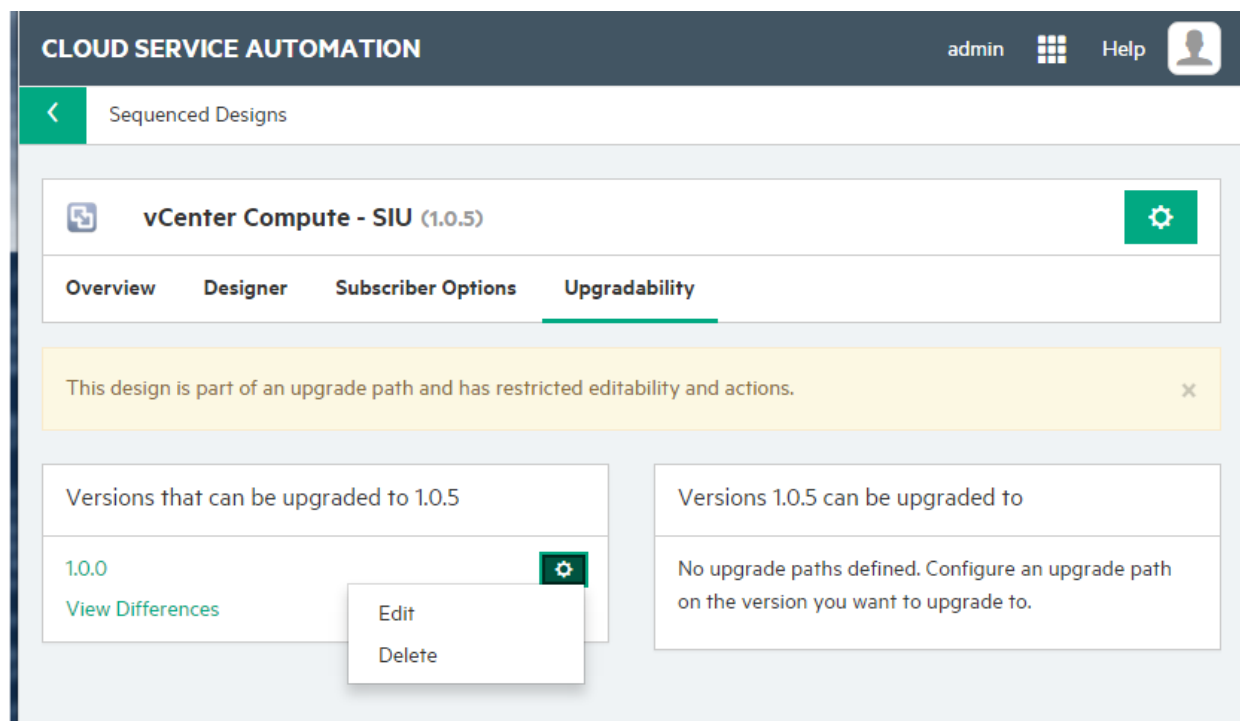
Figure 13. The **Upgradability** tab in the service design version page of the Sequenced Designs area



Edit upgrade path description or delete an upgrade path

In the **Upgradability** tab of the service design version page, on the left under **Versions that can be upgraded to <current version>**, click the gear icon and select to edit the upgrade path description of what has changed between the two designs or delete the upgrade path.

Figure 14. **Upgradability** tab options to edit the description of an upgrade path or delete an upgrade path

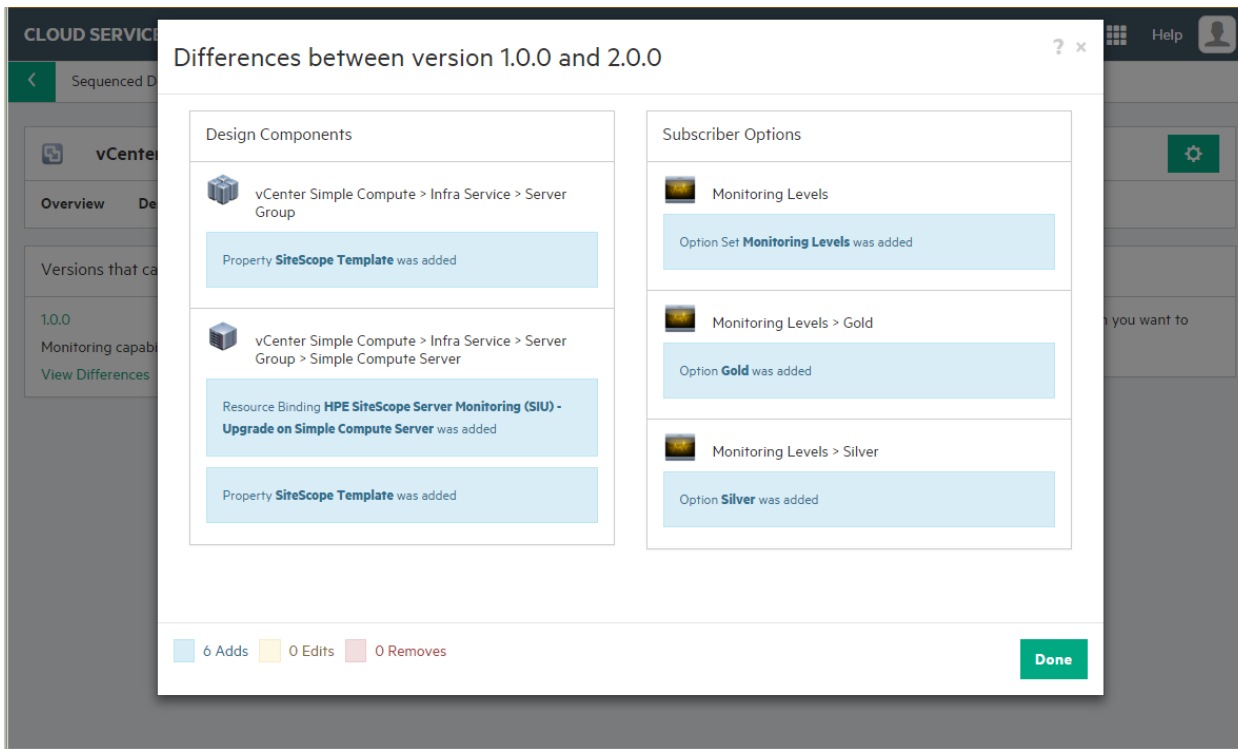


Deleting the upgrade path only removes the upgrade path between the two designs. It does not delete the upgrade service design. An upgrade path may not be deleted if an upgrade service offering that upgrades the prior design version to the current version exists. A warning will appear if such an attempt is made.

View differences between design versions

Upon establishing an upgrade path between different versions of service designs, changes made to the upgrade service design are viewable. In the **Upgradability** tab of the service design version page, on the left under **Versions that can be upgraded to <current version>**, click **View Differences**.

Figure 15. Changes made to the upgrade service design compared to the upgradable service design



Configure a service offering

The second step to upgrade a service instance is to configure an upgrade service offering:

1. Create an upgrade service offering.
2. Edit an upgrade service offering.

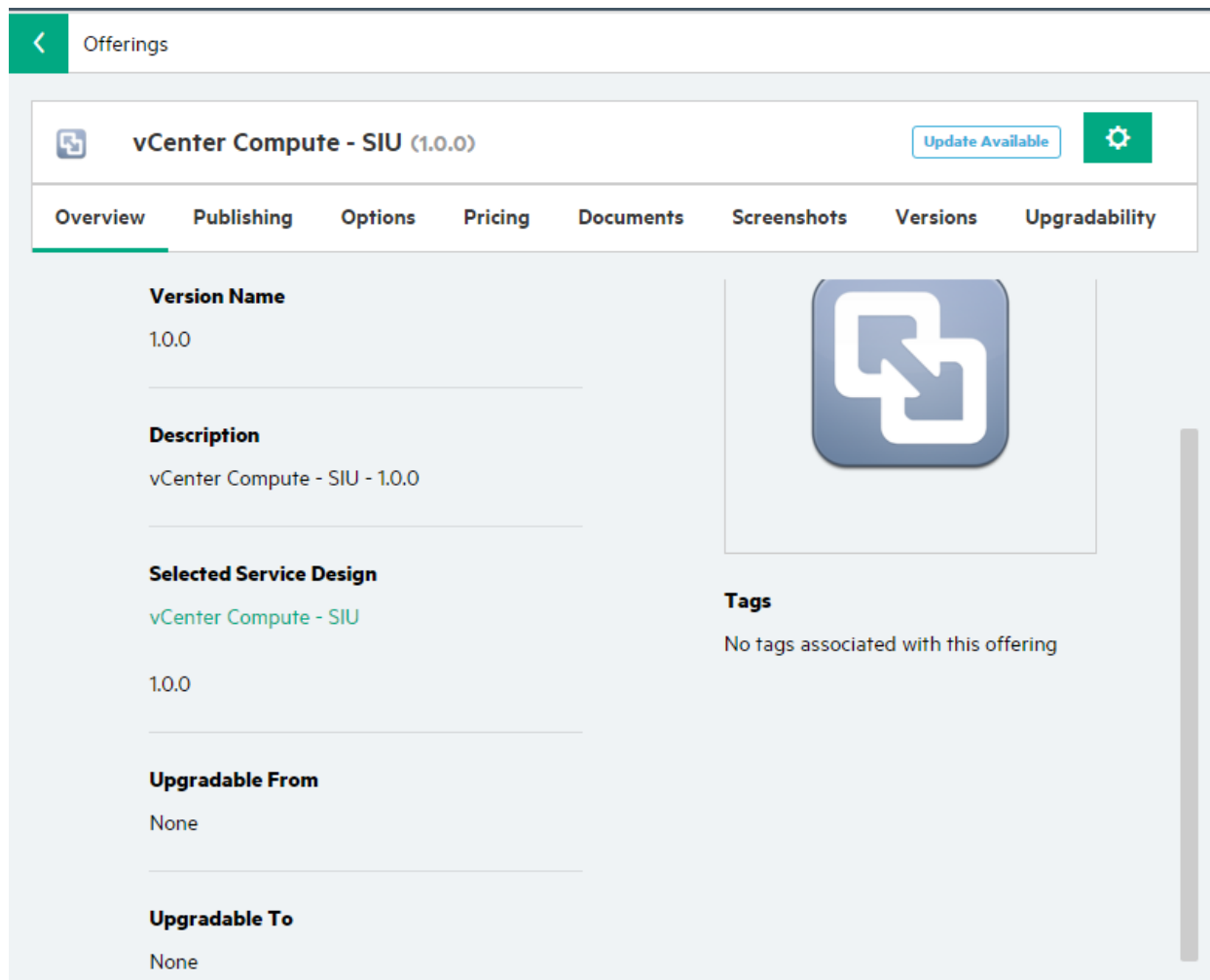
Additional tasks for service offering upgrades include the following:

- View upgrade paths.
- Edit upgrade path description or delete an upgrade path.

Create an upgrade service offering

After an upgrade service design has been created and published, service offerings created from the original (upgradable) service design have an **Update Available** label visible in the top right corner for a selected offering. This indicates that an upgrade service offering can be created from the original service offering.

Figure 16. **Update Available** label indicating that an upgrade offering can be created from this service offering



There are two ways to create an upgrade service offering for upgrading service instances:

- 1. Create a new service offering version
- 2. Establish an upgrade path between two offering versions

Once a service offering has an upgrade path established to a new upgrade version, an **Upgradable** label appears.

Figure 17. Service offering version that has an available upgrade

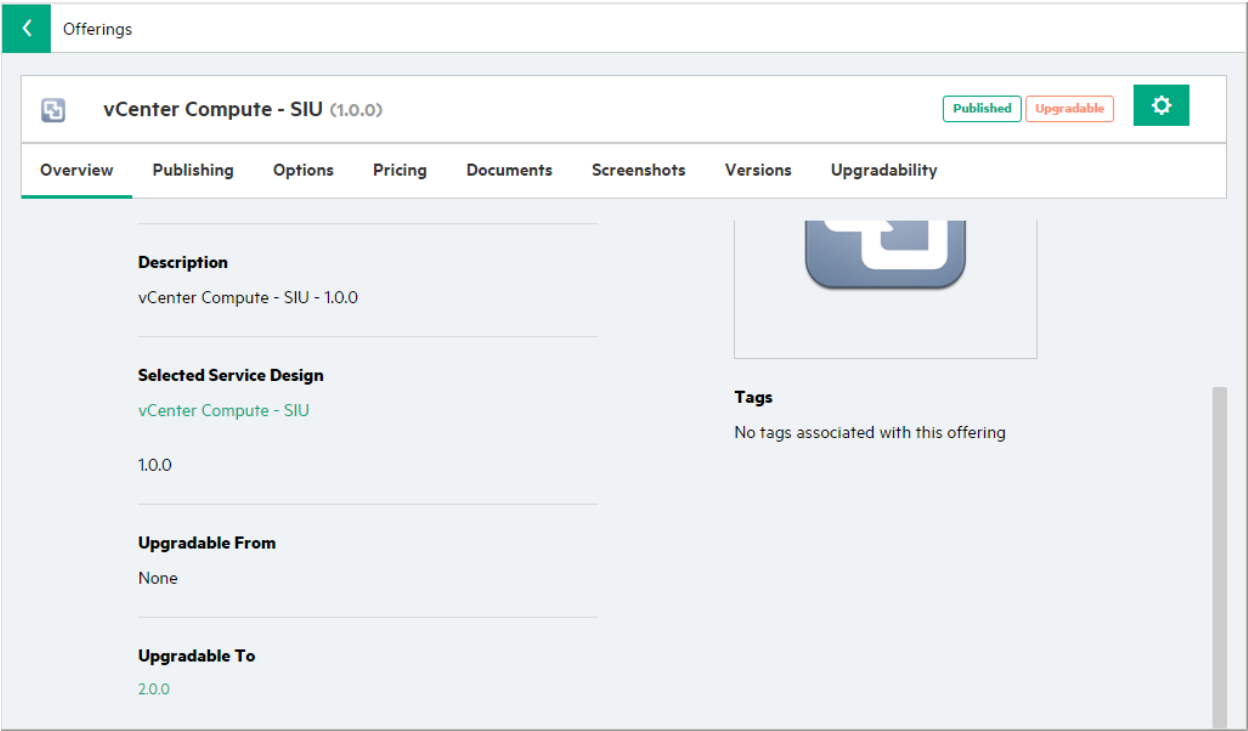


Figure 18. Service offering version list that shows an available upgrade

Offering Versions
All versions of Offering.

Offering (Version)	Description	Design (Version)	
vCenter Compute - SIU 2.0.0	vCenter Compute - SIU	vCenter Compute - SIU 2.0.0	
vCenter Compute - SIU 1.0.0	vCenter Compute - SIU - 1.0.0	vCenter Compute - SIU 1.0.0	<div>PublishedUpgradable</div>

Create a new service offering version

To create a new service offering version from an existing version:

1. Open the existing offering, the offering for which you wish to create an upgrade offering. This offering display the **Update Available** label.
2. Click the gear icon and select **Create New Version**.
3. Enter a new version (required), select **Yes** for **Upgradable from <selected version>**, and select the upgrade service design. Click **View Service Design Upgrade Description** to display the description of the upgrade path. This link displays only if the upgrade description is already defined in the designer.

The screenshot shows a 'New Version' dialog box with the following fields and options:

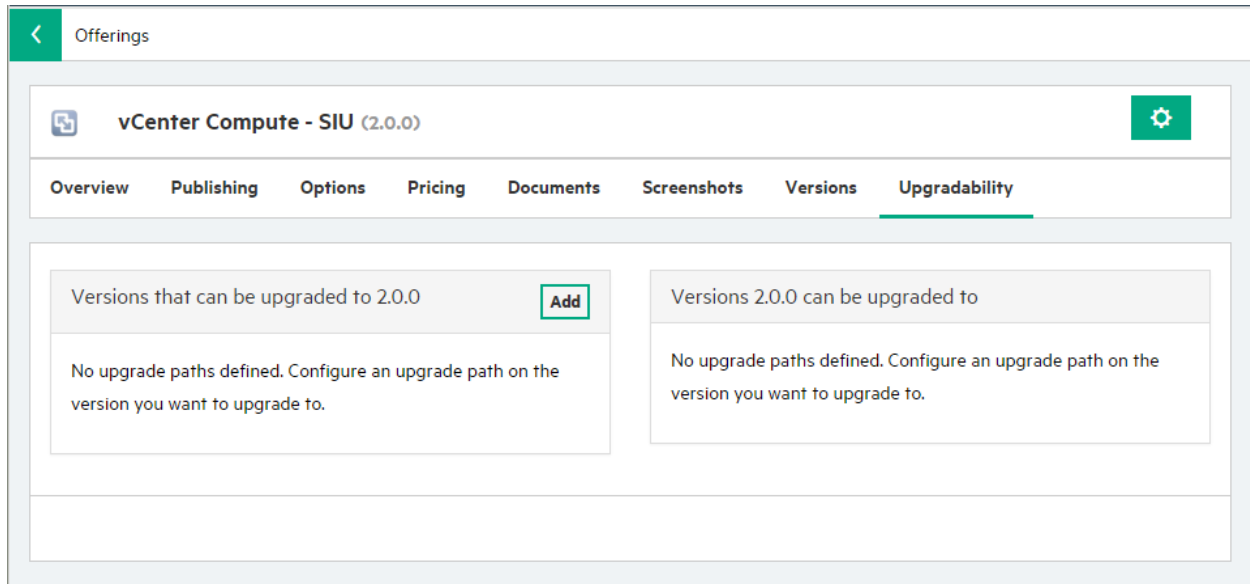
- Display Name:** vCenter Compute - SIU
- Offering Version ***: 1.0.0
- Upgradable from 1.0.0:** Yes (dropdown menu)
- Service Design ***: vCenter Compute - SIU (2.0.0) (dropdown menu with a menu icon and a help icon)
- Image:** A blue square icon with a white arrow pointing to the top-left corner. A **Change Image** button is next to it. Below the image, it says: Recommended dimension of 256x256. Maximum file size of 1MB.
- Tags:** No tags associated with this offering. A **Select Tags ?** link is next to it.
- Service Design 1.0.0 to 2.0.0 Upgrade:** A link **View Service Design Upgrade Description**.
- Description:** A large empty text area.
- Buttons:** **Create** and **Cancel** buttons at the bottom right.

4. Click **Create**. This creates an upgrade offering and establishes an upgrade path or relationship between the two offerings.

Establish an upgrade path between two offering versions

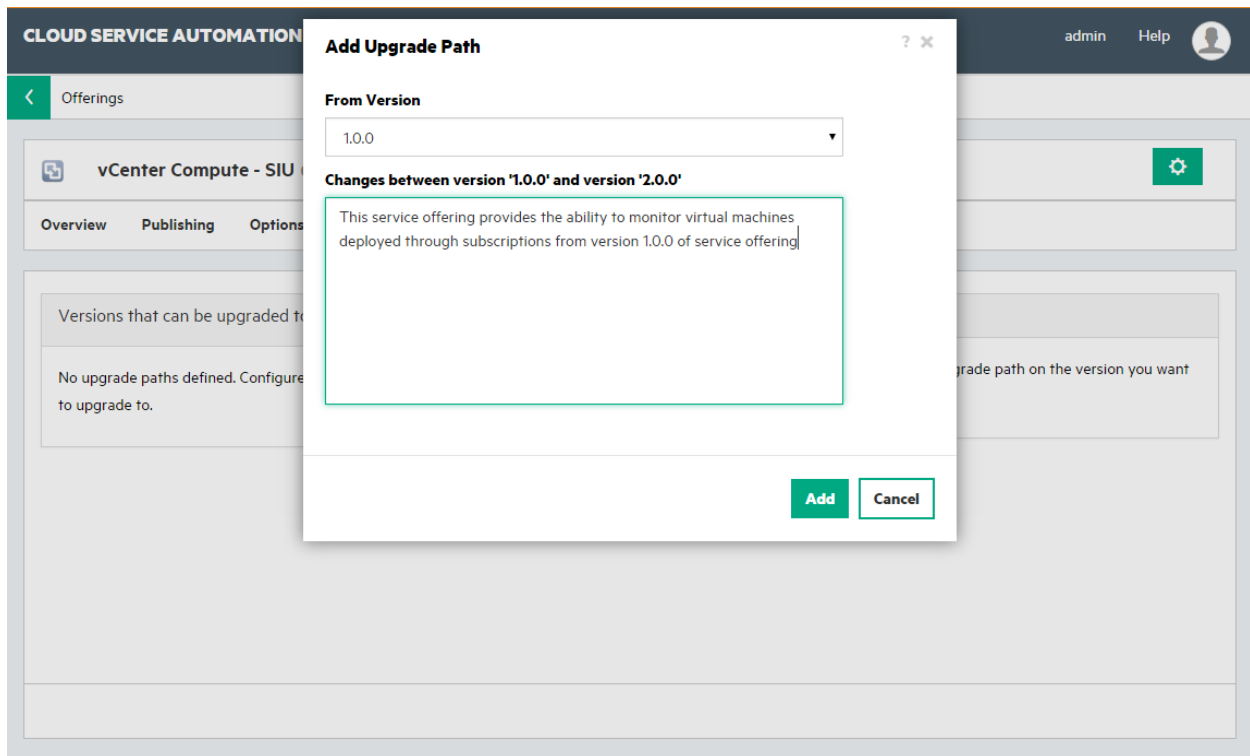
The second way to create an upgrade service offering is to establish an upgrade path or relationship between two versions of a service offering:

1. Open the offering that is to be the upgrade service offering. The selected version of the service offering must be cloned from the existing offering from which the upgrade path will be established.
2. Select the upgradability tab and click Add.



Offering versions which can act as upgradable offerings can be selected in the dialog.

3. Select an offering version and add a description of the changes between the two offerings.



4. Click Add. This establishes an upgrade path between the two offerings.

Note: Establishing an upgrade path between offering versions using these steps does not restore subscriber options and pricing changes made to the upgradable offering.

Edit an upgrade service offering

An upgrade service offering created from an existing version of a service offering has pricing changes and option model changes preserved from the previous version of service offering. New prices must be added and options model customizations must be completed for any new option sets, options or option properties added to the upgrade service design.

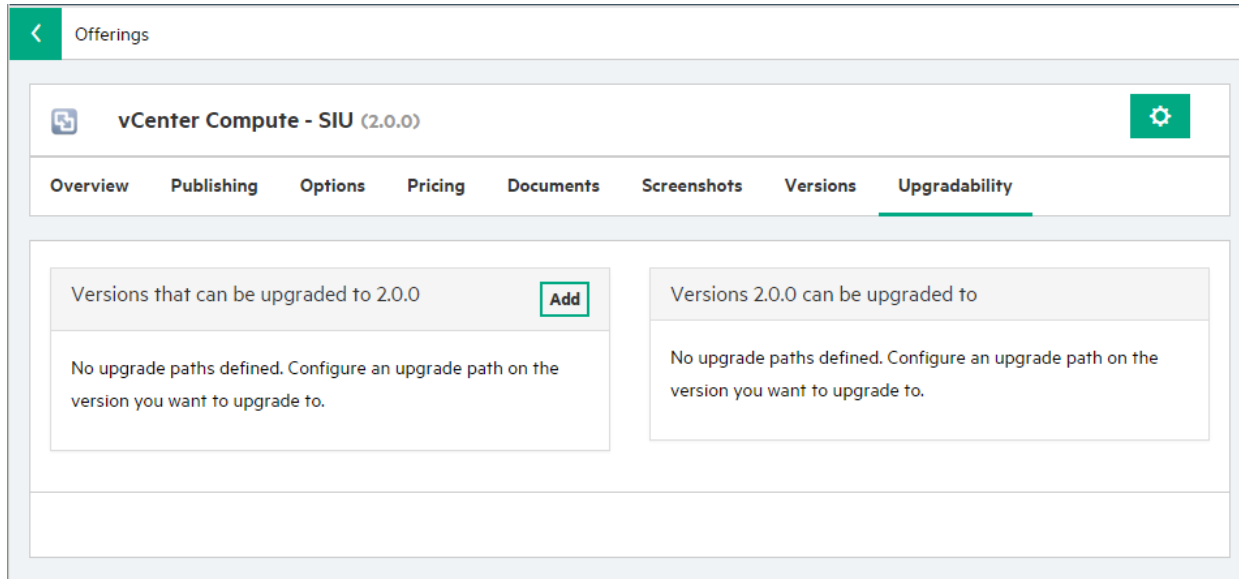
Note: If the original service offering has a customized display name, a change in the display name of an option set, option or option property of an upgrade service design is not reflected in the upgrade service offering. Service offering subscriber option customizations are given precedence over service design subscriber option changes.

Note: Creating an upgrade service offering from an existing service offering preserves pricing from the original service offering. Additional pricing changes, such as modifying existing prices and adding new prices to newly added options or option properties, can be made after creating the upgrade service offering. Only the Service Operations Manager can upgrade service subscriptions, therefore a subscriber will be unaware of any cost change until after the upgrade process has started.

View upgrade paths

The **Upgradability** tab displays the versions of service offerings that can be upgraded to the current version as well as the versions of service offerings to which the current version can upgrade.

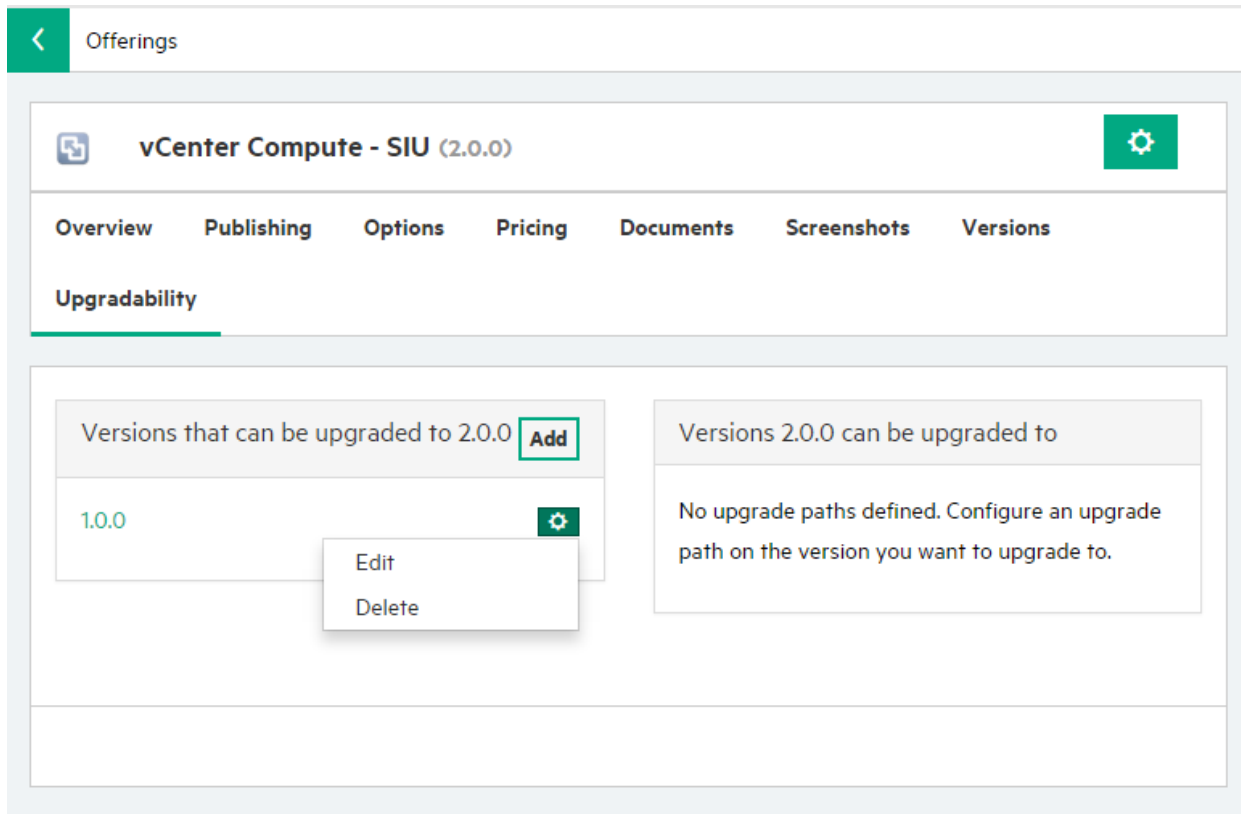
Figure 19. The **Upgradability** tab in the service offerings version page of the Offerings area



Edit upgrade path description or delete an upgrade path

In the **Upgradability** tab of the service offering version page, on the left under **Versions that can be upgrade to <current version>**, click the gear icon and select to edit the upgrade path description of what has changed between the two offerings or to delete the upgrade path. The Service Operations Manager uses the description to understand what has changed in the upgrade offering before upgrading service instances.

Figure 20. **Upgradability** tab options to edit the description of an upgrade path or delete an upgrade path



Deleting the upgrade path only removes the upgrade path between the two offerings. It does not delete the upgrade service offering.

Publish an upgrade service offering

The third step to upgrade a service instance is to publish the upgrade service offering. The upgrade service offering must be published into the same catalog as the upgradable service offering from which the service subscriptions were created.

Note: If the upgrade service offering is not published, the service subscription cannot be upgraded.

Upgrade a subscription

The Operations area of the Cloud Service Management Console displays service subscriptions by organization and, depending on the tab selected, subscriptions by users or service offerings.

Select a user or offering to view a list of service subscriptions. A service subscription that has an upgrade available is clearly indicated through an **Upgrade Available** label.

Figure 21. Subscription list displaying that upgrades are available for service subscriptions

Operations

consumer (consumer@csaconsumer.com)

4

0

0

0

0













0

Subscriptions

Subscriptions

Submitted On

Search

Subscription Details	Submitted On	Subscription Period	Subscription Status	Service Instance Status
 My Virtual Machines for Building Code Simple Compute Linux (Global Shared Catalog) Upgrade Available	01/12/2016 12:09:05 AM	01/11/2016 - no end date	 Active	 Online
 My Virtual Machines for Test Automation Simple Compute Linux (Global Shared Catalog) Upgrade Available	01/12/2016 12:08:39 AM	01/11/2016 - no end date	 Active	 Online
 My Virtual Machines for Development Simple Compute Linux (Global Shared Catalog) Upgrade Available	01/12/2016 12:08:17 AM	01/11/2016 - no end date	 Active	 Online
 Simple Compute Linux (1.0.0) Simple Compute Linux (Global Shared Catalog) Upgrade Available	01/06/2016 6:04:18 PM	01/06/2016 - no end date	 Active	 Online

4 Total Items

Selecting a service subscription that has an upgrade available displays additional information, such as available upgrade versions.

Figure 22. Service subscription view displaying available upgrade versions

Operations

My Virtual Machines for Development

Upgrade Available

Active

Online

Overview

Events

Topology

Providers

Display Name

My Virtual Machines for Development

Service Offering (Service Catalog)

Simple Compute Linux 1.0.0 (Global Shared Catalog)

Design Name

Simple Compute Linux (1.0.0)

Subscriber

consumer (consumer@csaconsumer.com)

Submitted On

01/12/2016 12:08:17 AM

Subscription Period

01/11/2016 - no end date

Subscription Cost

\$200.00 and \$100.00 Yearly

Subscription Status

Active

Service Instance Status

Online

Upgradable to

2.0.0

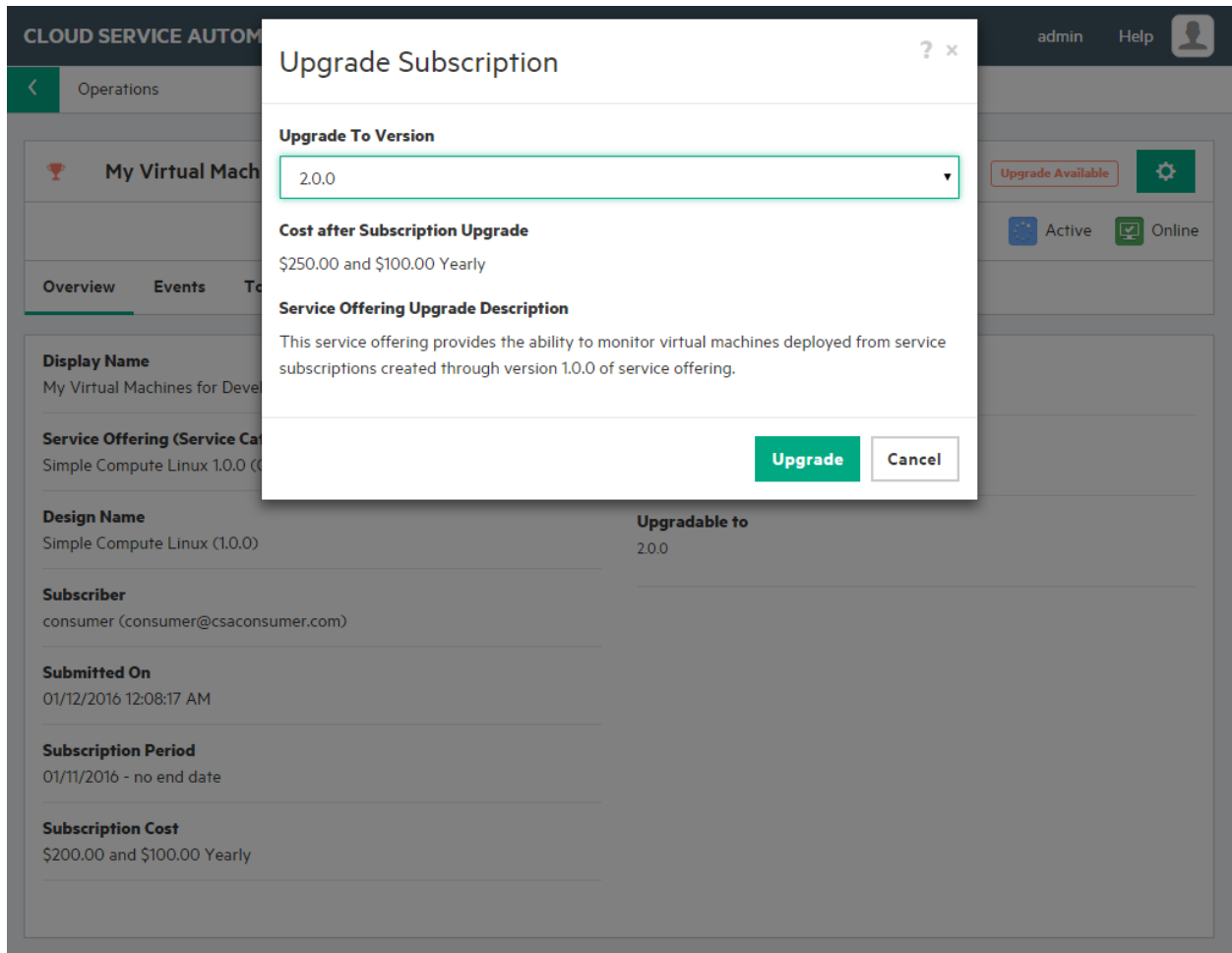
To upgrade a subscription:

1. Click the gear icon and select **Upgrade**.

The screenshot shows the 'Operations' console interface. At the top, there's a header with a back arrow and the word 'Operations'. Below this is a section titled 'My Virtual Machines for Development' with a red 'Upgrade Available' button and a gear icon. A context menu is open over the gear icon, showing options: 'Upgrade', 'Transfer', 'Cancel', and 'Refresh'. The 'Upgrade' option is highlighted. Below the header, there are tabs for 'Overview', 'Events', 'Topology', and 'Providers'. The 'Overview' tab is selected, displaying subscription details in a two-column layout.

Display Name	Subscription Status
My Virtual Machines for Development	Active
Service Offering (Service Catalog)	Service Instance Status
Simple Compute Linux 1.0.0 (Global Shared Catalog)	Online
Design Name	Upgradable to
Simple Compute Linux (1.0.0)	2.0.0
Subscriber	
consumer (consumer@csaconsumer.com)	
Submitted On	
01/12/2016 12:08:17 AM	
Subscription Period	
01/11/2016 - no end date	
Subscription Cost	
\$200.00 and \$100.00 Yearly	

2. Select the version of the service offering to which to upgrade. The Upgrade Subscriptions dialog displays the cost of the service subscription after upgrade and a description of the upgrade service offering.



- Click Upgrade. The service subscription status changes from Active to Pending and then the service instance state changes from Online to Upgrading. During this process, the service subscription displays the new service offering selected for upgrade as the offering from which the selected service offering originated.

The screenshot shows a web interface for managing virtual machines. At the top, there's a header with a back arrow and the word 'Operations'. Below that, a title bar reads 'My Virtual Machines for Development' with a settings gear icon on the right. A status bar indicates 'Pending' and 'Upgrading' with corresponding icons. A navigation menu below the title bar includes 'Overview' (selected), 'Events', 'Topology', and 'Providers'. The main content area is divided into two columns. The left column lists subscription details: Display Name (My Virtual Machines for Development), Service Offering (Simple Compute Linux 2.0.0 (Global Shared Catalog)), Design Name (Simple Compute Linux (2.0.0)), Subscriber (consumer (consumer@csaconsumer.com)), Submitted On (01/12/2016 12:08:17 AM), Subscription Period (01/11/2016 - no end date), and Subscription Cost (\$250.00 and \$100.00 Yearly). The right column shows the Subscription Status (Pending), Service Instance Status (Upgrading), and an 'Upgradable to' section stating 'No upgrade paths defined.'

About the service subscription upgrade process

- When a service subscription upgrade is attempted, the following validations are performed prior to submitting the service request that initiates the upgrade:
 - Subscription status is Active
 - Service Instance state is Online or Public Action Failed
 - Subscription does not have pending requests (such as cancel, modify or upgrade subscriptions)
 - Service offerings (upgradable, upgrade) are published, from the same catalog, and an upgrade path exists between them
 - Service designs (upgradable, upgrade) are published and an upgrade path exists between them
- The subscriber is notified about various events of the subscription upgrade process such as the start of the upgrade process, completion of the upgrade, failure during the upgrade and subscription cost change during the upgrade.
- The subscription reflects the selected upgrade service offering version as the service offering underlying the subscription. If the subscription upgrade results in a new cost for the subscription, the new cost is reflected for the service subscription.
- If the subscriber selected certain options during the creation or modification of the subscription, these selections are always preserved. New options or changes in the default option selection in the upgrade service offering do not affect the selected options.
- If the subscriber entered values for subscriber option properties during subscription creation or modification, these values are preserved in the service instance component properties after subscription creation or modification. Such values are always preserved after the subscription upgrade process, both in subscription option properties and also in the service instance component properties.

- During the creation of an upgrade service design, values of subscriber option properties that are not user editable could have changed. Such changes in subscription option property values are reflected in the service instance component properties after service instance has been upgraded.

Behavior of Dynamic Properties during Service Instance Upgrade

This section provides details on how the SIU feature behaves when there are existing subscriber option properties that are dynamic and depend on the Portal /Client tokens (and) when new dynamic properties are added to the upgraded design.

- The property having dynamic JSPs in the Subscriber options of the Initial design

During an upgrade, the JS/JSP will be executed initially during the calculation of the price and later during the upgrade process.

The client tokens will be resolved properly and sent to the JS/JSP when the upgrade transition is triggered.

Portal tokens will not be resolved (if needed the JS/JSP will have to be modified so that it will return a default set of values or empty values) .They need not be the one that were selected during the ordering of the subscription from the Marketplace Portal for the first time and can be a default value that is to be used if the incoming portal token is not resolved in the JS/JSP.

Dynamic option scripts are executed to ensure the calculation of any change in the pricing for the selected option. If the selected option is unavailable during the execution of the JS/JSP, then the price will not change and upgrade will continue with the previously selected values.

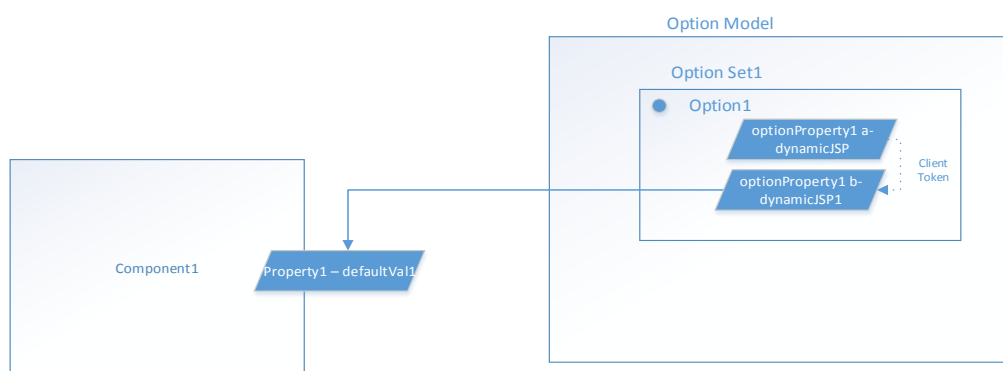
- A new dynamic list property added to the selected option within an option set (or) a new dynamic property added to the option set (or) existing static properties converted to dynamic property.

The associated JS/JSP for the new dynamic properties will not be executed. The default value of the property on the component to which the newly created dynamic property in subscriber options, is target bound, will be taken during the upgrade. Since the JS/JSP is not executed, the pricing will not be accounted for and is currently the limitation. Only after doing a modification from the portal side to select the specific JS/JSP value on the upgraded subscription, the price will be added.

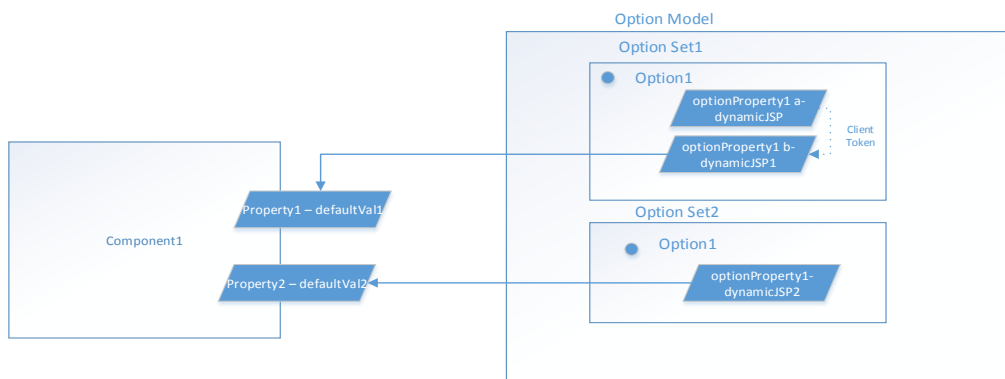
Examples to illustrate this:

Scenario 1: A new dynamic property added in the upgraded design

Design 1.0:



Design 2.0:

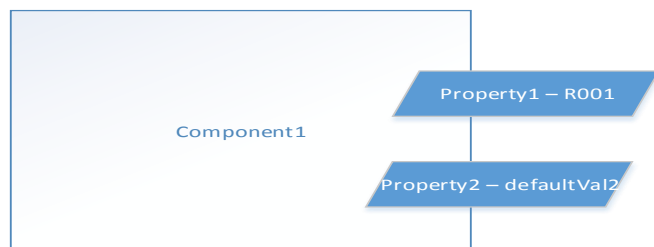


The design 1.0 has a dynamic property (optionProperty1b) that has a JSP that takes a client token. The initial subscription is created from design 1.0 and the Property1 takes the value R001. In the upgraded design, there is a new dynamic property (tied to dynamicJSP2) in the subscriber option. It is target bound to a property in Component1 that has a default value – defaultVal2.

Expected behavior on upgrade:

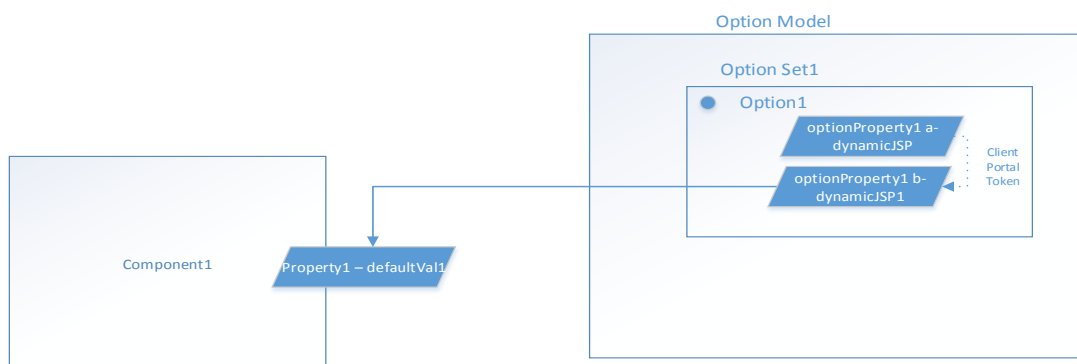
- DynamicJSP1 will be executed and the client token will be resolved. The updated value of price, if any, for the option is updated.
- DynamicJSP2 will not be executed. The upgrade will proceed with the value defaultVal2 for the Property2 of Component1.

Result:

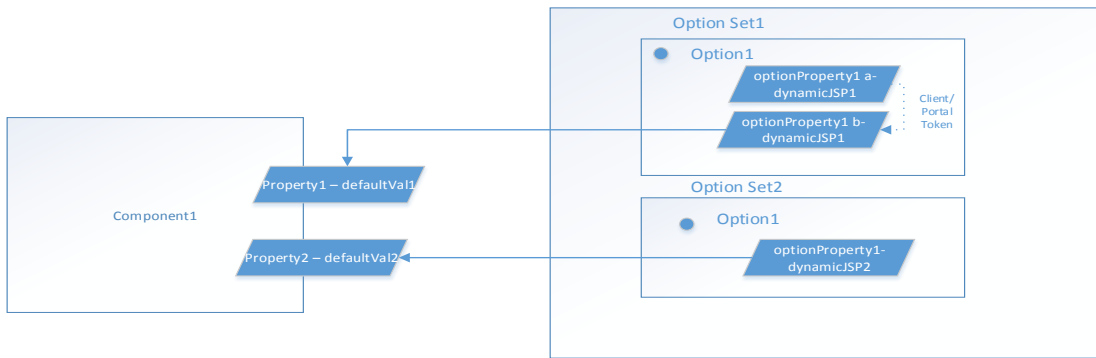


Scenario 2: Exiting Design has Portal Tokens

Design 1.0:



Design 2.0:

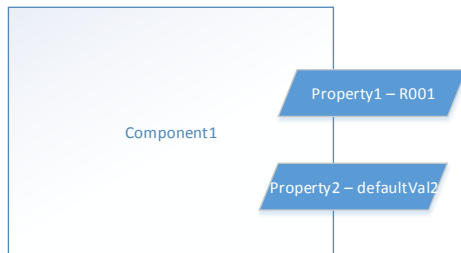


The initial design has Client and Portal token for dynamicJSP1. The initial subscription created from MPP has the values set to R001 for Property1 of Component1.

In the upgraded design, there is a new dynamic property (tied to dynamicJSP2) in the subscriber option. It is target bound to a property in Component1 that has a default value – defaultVal2.

Expected Behavior:

- DynamicJSP1 will be executed. Client tokens are resolved. But Portal Tokens were not resolved and DynamicJSP1 returned empty values.
In such case - The originally selected values R001 will be selected and upgrade will continue with the value of R001 for Property1 for Component1.
- DynamicJSP2 will not be executed. The upgrade will proceed with the value defaultVal2 for the Property2 of Component1.
- Result:

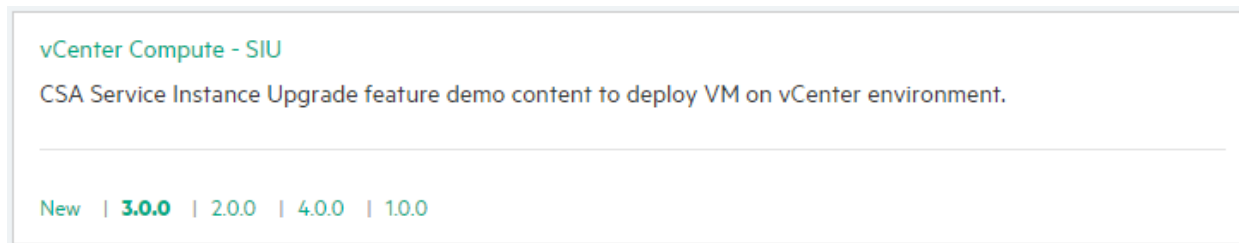


Use cases for constructing upgrade service designs

CSA ships with sample designs to demonstrate use cases for upgrading service instances. This section discusses upgrade use cases which include changes added to new versions of service designs. We will also discuss how to incrementally add features to service instances by upgrading them from one version of a service design to another.

The service design we will be using for discussion is named **vCenter Compute – SIU**. This design has four versions: 1.0.0, 2.0.0, 3.0.0 and 4.0.0. Version 2.0.0 is an upgrade for version 1.0.0, version 3.0.0 is an upgrade for version 2.0.0 and version 4.0.0 is an upgrade for version 3.0.0.

Figure 23. Sample service design to demonstrate upgrading service instances



Note: These service designs were cloned from one another but do not have an upgradability relationship established. An upgrade relationship needs to be established between the higher version and lower version of the service design from the higher version of service design (see Establish an upgrade path between two design versions for more information). The lower version of the service design should be published before establishing this relationship.

Version 1.0.0 of this service design deploys virtual machines using the VMWare vCenter resource provider. A VMWare vCenter resource offering attached to the server component has OO flows defined to deploy the server. There are three compute options (small, medium and large) and option properties (Number of CPUs and Memory) that are tied to corresponding server component properties. Server components from the service instances deployed through this service design have public actions to execute (Start Server, Stop Server, Restart Server and Suspend Server).

Use case 1 – add monitoring capability using HPE SiteScope to server components in service instances

To provide additional capability such as monitoring to existing virtual machines during an instance upgrade involves the following changes to the upgrade service design:

- Add a new resource offering to existing server components that includes provider selection actions in the 'Upgrading Pre-Transition' state. Then add actions to deploy monitors in the 'Upgrading Transition' lifecycle states. Similar provider selection actions for resource bindings need to be added to the resource offering in the 'Deploying' lifecycle state for deploying monitors to virtual machines deployed through the new version of the service design.
- Add a new component property that specifies the monitor to be applied to the virtual machines.
- Add an option set with one or more options where the options have a property with a default value bound to the component property that specifies the monitor to be applied. One of the options needs to be the default option that is applied for upgrading service instances.

Version 2.0.0 of the **vCenter Compute – SIU** service design provided with CSA is an upgrade for version 1.0.0. It provides monitoring capability (using HPE SiteScope) to virtual machines either deployed from version 1.0.0 of the service design through an upgrade process or to virtual machines deployed through service instances originating from version 2.0.0.

A new resource binding to the HPE SiteScope (SiteScope) resource offering has been added to the **Simple Compute Server** component in version 2.0.0 of the service design. The resource binding provides monitoring capability to virtual machines. As you will see from the resource binding, there are **Build Resource Provider List** and **Select Resource Provider** actions configured in two different locations. The first is located under the **Provider Selection** tab of the resource binding and the second is located under the **Upgrading Pre-Transition** lifecycle state of the resource offering. The first location is used for SiteScope provider selection when deploying new service instances from version 2.0.0 of the service design. Actions configured in the **Upgrading Pre-Transition** lifecycle state are used for SiteScope provider selection when upgrading service instances from version 1.0.0 of the service design. Similarly, you will also observe that a **Simple Compute Monitor – Deploy** action is configured in the **Deploying Transition** and **Upgrading Transition** lifecycle states. The action in the **Deploying Transition** lifecycle state deploys the SiteScope monitor to virtual machines based on virtual machines deployed directly from version 2.0.0 of the service design. The action in the **Upgrading Transition** lifecycle state deploys the SiteScope monitor to virtual machines when upgrading service instances from version 1.0.0 of the service design.

Other changes you will observe in version 2.0.0 of the service design include a new **sitescopeTemplate** property on the **Simple Compute Server** component which is utilized for passing an appropriate SiteScope template value to an OO flow deploying the template. This service design also has a new Monitoring Levels option set with two options (Gold and Silver) where **Sitescope Template Name** option property is bound to the **sitescopeTemplate** component property. Since Silver is the default option selected, at the end of upgrading service instances, the virtual machines have a Silver template applied for monitoring purposes. After the upgrade process, subscribers can modify their subscriptions to have the ability to select the Gold option for applying the Gold template for monitoring virtual machines using SiteScope.

Use case 2 – deploy applications using HPE Server Automation to server components in existing service instances

To add applications to existing virtual machines during an instance upgrade involves the following changes to the upgrade service design:

- Add one or more new service components that represent applications to upgrade service design under existing service components.
- Add resource bindings to application-based resource offerings for deploying applications.
- Add option sets where option properties from underlying options are bound to the new application component properties in the design.

Version 3.0.0 of the **vCenter Compute – SIU** service design provided with CSA is an upgrade for version 2.0.0. It deploys applications using HPE Server Automation (SA) to virtual machines deployed from version 2.0.0 of the service design through the upgrade process. This design version can also be used for provisioning applications by creating subscriptions using this design.

Note: Since service instances from version 1.0.0 of the service design can be upgraded to version 2.0.0 of service design and service instances from version 2.0.0 of the service design can be upgraded to version 3.0.0 of the service design, the service instance upgrade process can be utilized for rolling out services in an incremental manner.

Note: In version 3.0.0 of the service design, the existing binding to the SiteScope resource offering does not have any actions in the Upgrading lifecycle state as compared to version 2.0.0. Since service instances representing design version 2.0.0 already have SiteScope monitoring deployed, on upgrade from 2.0.0 to 3.0.0, it is not intended to run the SiteScope-based OO flows again. Since deletion of resource bindings is not allowed on upgrade service designs, resource offering replacement functionality has been utilized to replace an existing resource offering with another resource offering that does not have actions in the Upgrading lifecycle state.

In version 3.0.0 of the service design, a new resource binding to an SA resource offering has been added to the **Simple Compute Server** component for installing an SA agent to deployed virtual machines. This resource binding creates instances from this design or by upgrading instances from version 2.0.0 of the service design. Actions have been added to the Upgrading and Deploying lifecycle states to support SA agent installation when upgrading service instances from version 2.0.0 and also to support direct deployment.

Additionally a new service component, **Software (chef solo)**, has been added to the **Simple Compute Server** component that represents the application component. A resource binding to a different SA resource offering has been added to this service component for deploying applications. Since **Software (chef solo)** is a new component and since actions in the Initializing, Reserving and Deploying lifecycle states are executed for this component, all the actions for the SA resource offering have been added to these lifecycle states.

A new option set with an option that has properties related to application deployment has been added to this service design and these option properties are bound to component properties. Also note that the option properties have default values so that they are used as the default for upgrading instances.

After upgrading service instances from version 2.0.0 of service design, applications are deployed to the service instances based on the upgrade design version 3.0.0.

Use case 3 – provide additional public actions to service instance components

This is a very simple use case and the steps to create the upgrade service design is to add additional public actions to existing service components or resource offerings bound to the service components in the Deployed lifecycle state.

Version 4.0.0 of the **vCenter Compute – SIU** service design provided with CSA is an upgrade for version 3.0.0. It provides additional public actions to the server-based components. A VMWare vCenter resource offering bound to the **Simple Compute Server** component has an additional public action named **Refresh Server Info** configured at the Deployed lifecycle state.

After the service instance upgrade, the server components display an additional public action to update server information.

Note: In version 4.0.0 of the service design, resource offerings bound to service components do not have actions in the Upgrading lifecycle state. Resource offerings with actions in the Upgrading lifecycle state (of service design version 3.0.0) were replaced with resource offerings without actions in the Upgrading lifecycle state. This was done so that during upgrade, actions present in the Upgrading lifecycle state in version 3.0.0 are not executed again. Actions configured in the Upgrading lifecycle state can also be re-

runnable (meaning that after upgrade, running these actions do not have any effect on the service instance) instead of replacing resource offerings.

Use case 4 – upgrade an application deployed through a sequenced design

CSA does not provide a sample for this use case. However, it can be achieved by adding actions to upgrade applications in the Upgrading lifecycle state. Inputs, such as application version, can be made available through an option model property bound to application components.

Use case 5 – configure additional firewall rules or apply patches to virtual machines deployed through a sequenced design.

Similar to use case 4, this use case can be achieved by configuring additional options and actions related to adding firewall rules or applying patches to the Upgrading lifecycle state. Inputs such as firewall ports or list of patches can be made available through an option model property bound to server components. Subscribers can also apply firewall rules or patches to virtual machines by themselves if additional actions related to adding new firewall rules or applying patches are added to Modifying lifecycle state.




Use case 6 – fix problems with modifying or cancelling an existing subscription

This use case fixes problems in an existing subscription by correcting, adding or removing actions from the Modifying, Un-deploying, Un-reserving or Un-initializing lifecycle states of components or resource offerings bound to the components.

Failure handling when upgrading a service instance

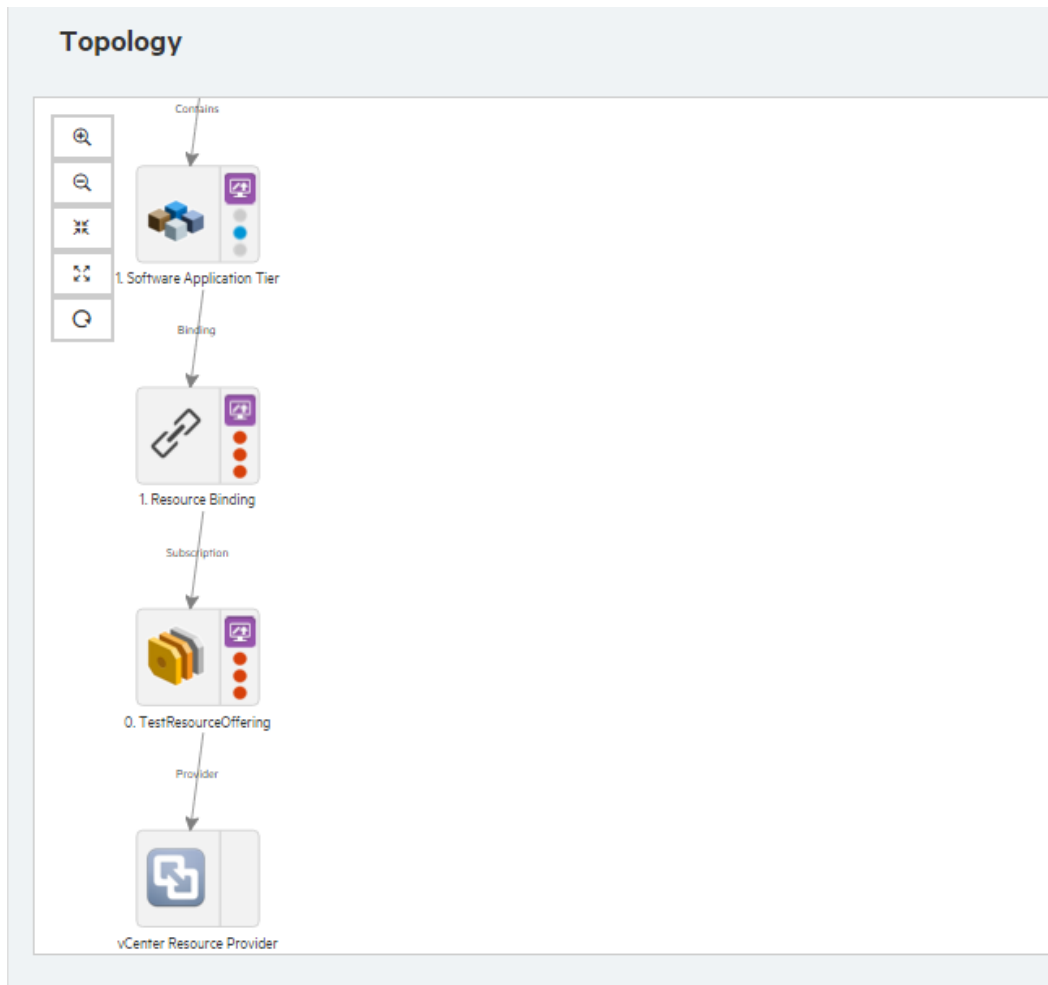
A service subscription upgrade can fail when an error occurs during the upgrade process. For example, an error could occur during the execution of an OO flow that is part of the Upgrading lifecycle state. When an error occurs during the upgrade process, the service subscription state remains Active, and the service instance state is set to Upgrade Failed.

Figure 24. When a subscription upgrade fails, the subscription status is Active and the instance status is Upgrade Failed

Subscriptions				
		Submitted On	Search	
Subscription Details	Submitted On	Subscription Period	Subscription Status	Service Instance Status
 ServiceInstanceUpgradeSample (1.0.0) ServiceInstanceUpgradeSample (Global Shared Catalog)	01/06/2016 3:01:30 PM	01/06/2016 - no end date	 Active	 Upgrade Failed

In the service subscription view of the Operations area in the Cloud Service Management Console, select the Topology tab to view the service instance topology and identify the node within the topology hierarchy that is the root cause of the error. In the following example, there is an error upgrading the resource offering.

Figure 25. Service instance topology view of an upgrade failure during the upgrade of a resource offering



When an upgrade failure occurs, the Service Operations Manager can work to correct the error condition. For example, the Service Operations Manager can correct the OO flow, and re-attempt the service subscription upgrade from the Operations area of the Cloud Service Management Console. The service subscription that failed to upgrade can also be cancelled either by the subscriber or the Service Operations Manager.

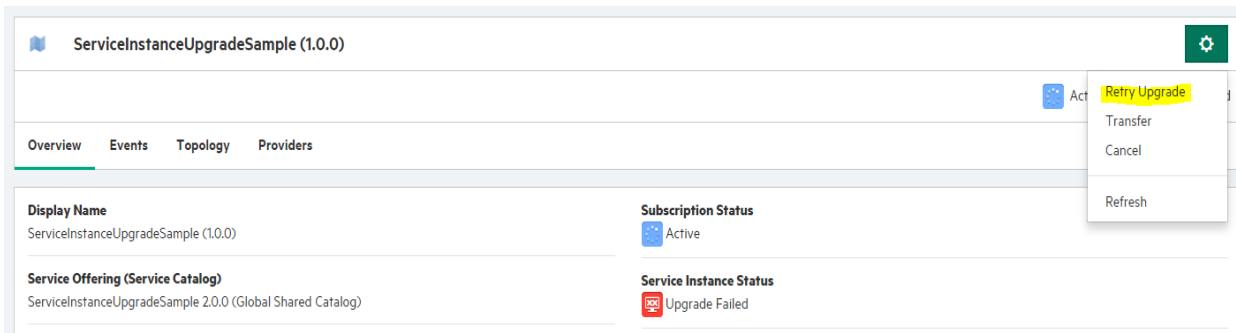
Note: When an upgrade failure occurs, the lifecycle transition stops at the action that failed during the upgrade process. When the subscription upgrade process is re-attempted, the lifecycle transition of the service instance continues where it stopped during the original upgrade process and the action that failed during the original upgrade process is re-executed.

Token to indicate upgrade retry

There is token that can be passed as a parameter to the OO flow called ACTION_EXECUTION_RETRY. If the OO flow supports that input parameter, then the parameter's input value will be set to true or false, depending upon if the action is being executed for the first time (ACTION_EXECUTION_RETRY = false), or if the action is being re-tried (ACTION_EXECUTION_RETRY = true). This allows the

OO flow to be written such that step(s) can be re-executed. There is no limit to the number of times a failed service subscription upgrade can be re-attempted. Once the failure is corrected, the subsequent service subscription upgrade should be successful.

Figure 26. Click the gear icon and select **Retry Upgrade** to retry the upgrade



REST API to upgrade a service subscription

The REST API for performing a single service subscription upgrade is an HTTP POST request:

`https://<FQDN or IP address of CSA host>:<CSA port>/csa/api/service/subscription/<service subscription id>/upgrade/<upgrade service offering id>`

For example:

`https://mycsasystem.mydomain.com:8444/csa/api/service/subscription/90cefa215212e353015214027a9b1c15/upgrade/90cefa215212e35301521401da251ba4`

The JSON response contains the service request id:

```
{
  "id" : "90cefa215212e3530152140316f91c5a"
}
```

Upgrading multiple subscriptions originating from the same service offering

The REST API mentioned in the previous section can be used with other CSA REST API(s) to upgrade multiple or all service subscriptions from the same service offering (sometimes referred to as bulk upgrading).

In the Operations area of the Cloud Service Management Console, subscriptions can be filtered by organization. Within a selected organization, the subscriptions can be filtered by users or by the service offering on which the subscriptions are based. REST APIs can also be used to filter subscriptions in this manner. For example, if you wanted to upgrade a service offering for a specific upgradable version for which multiple outstanding active subscriptions exist across multiple users, do the following:

1. Retrieve the subscription(s) that are based on an upgradable service offering using an HTTP GET request:

`https://<FQDN or IP address of CSA host>:<CSA port>/csa/api/service/subscription/organization/<organization id>/offering/<upgradable service offering id>`

For each subscription that is based on the specified upgradable service offering within the specified organization, the JSON response contains the service subscription id, service subscription status, and if an upgrade is available. For example:

```
{
  ...
  "@self" : "/csa/api/service/subscription/90cefa215218184601521856d2c10089",
  ...
  "csa_subscription_status" : "ACTIVE",
  ...
  "csa_upgrade_available" : true,
  ...
}
```

2. Retrieve the upgrade service offering id for each service subscription using an HTTP GET request:

`https://<FQDN or IP address of CSA host>:<CSA port>/csa/api/service/subscription/<service subscription id>/upgrade-offerings`

For example:

`https://mycsasystem.mydomain.com:8444/csa/api/service/subscription/90cefa215218184601521856d2c10089/upgrade-offerings`

The JSON response contains the upgrade service offering id and version:

```
{
  ...
  "upgrades_from" : [ {
    "upgrade_id" : "90cefa215218184601521850b8560038",
    ...
    "upgrade_version" : "2.0.0"
  } ]}
  ...
}
```

3. Upgrade multiple service subscriptions based on an upgradable service offering to an upgrade service offering using the REST API mentioned in the previous section (an HTTP POST request) in a loop. An example of the REST API:

`https://mycsasystem.mydomain.com:8444/csa/api/service/subscription/90cefa215218184601521856d2c10089/upgrade/90cefa215218184601521850b8560038`

Send documentation feedback

If you have comments about this document, you can send them to clouddocs@hpe.com.

Legal notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted rights legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright notice

© Copyright 2017 Hewlett Packard Enterprise Development Company, L.P

Trademark notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation in the United States and other countries, and are used with the OpenStack Foundation's permission.

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to the following URL and sign-in or register: <https://softwaresupport.hpe.com>.

Select Manuals from the Dashboard menu to view all available documentation. Use the search and filter functions to find documentation, whitepapers, and other information sources.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your Hewlett Packard Enterprise sales representative for details.

Support

Visit the Hewlett Packard Enterprise Software Support Online web site at <https://softwaresupport.hpe.com>.