Cloud Service Automation

Resource Management

Software version: 4.50 and 4.60

Document release date: July 2016

Software release date: July 2016

Contents

Introduction

Cloud Service Automation (CSA) is not a management platform. However, CSA provides a model to represent various categories of resource providers, capacities, and utilization.

A resource provider is a lightweight representation of an external resource-management platform, such as a vCenter server or SiteScope server. CSA uses instances of resource providers to provision, retire, or operate a resource.

Some external resource-management platforms provide capabilities to manage their own resources. For example, a vCenter server manages the CPU and memory it provides as compute resources and a SiteScope server monitors licenses on a provisioned server. You can define a resource pool for the CSA provider instance. A resource pool models the capacity and utilization of resources that a provider instance manages. You can also configure and report resource-pool capacity values, and synchronize capacities between the management platform and the resource pool in CSA.

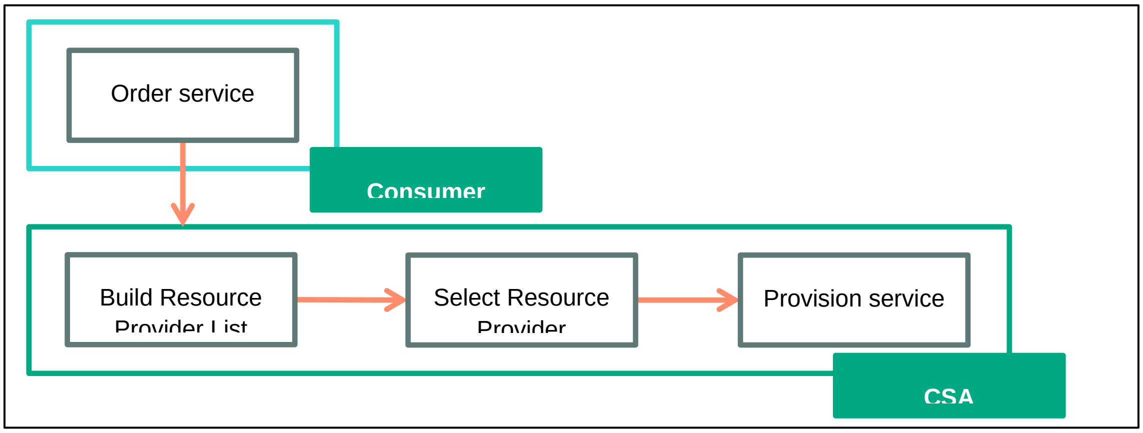
Resource management concepts

Note: CSA supports two design models for developing orderable services: sequenced and topology. Sequenced designs are currently the most common implementation for enterprise customers and, unless otherwise specified, are the models used for this document.

Provider selection

During the reservation phase of the service lifecycle, consumers order services and select providers. Subsequently, CSA uses internal actions (configured on a resource binding) to create a list of available resource providers capable of fulfilling the requirements of a resource-provisioning request.

Figure 1. Provider Selection



CSA performs the following actions specified in the Resource Binding Provider Selection action list:

- Build Resource Provider list action, which builds a list of valid resource providers that meet the following requirements:
    - Are capable of providing the resource offering that is referenced in a resource binding.
    - Belong to at least one resource environment associated with the corresponding service catalog, if the catalog has any configured resource environments. CSA creates service offerings from a service design and publishes the offerings in service catalogs.
    - Have their availability option set to **Enabled**.
- Import the Operations Orchestration (OO) filter flow to further refine the provider list.
- Select Resource Provider action, which selects a resource provider (at random if there is more than one provider) from the valid list of providers that the Build Resource Provider List created and the OO filter flow refined. After you specify a unique property name, CSA subsequently writes the selected provider to the property on the associated service component.

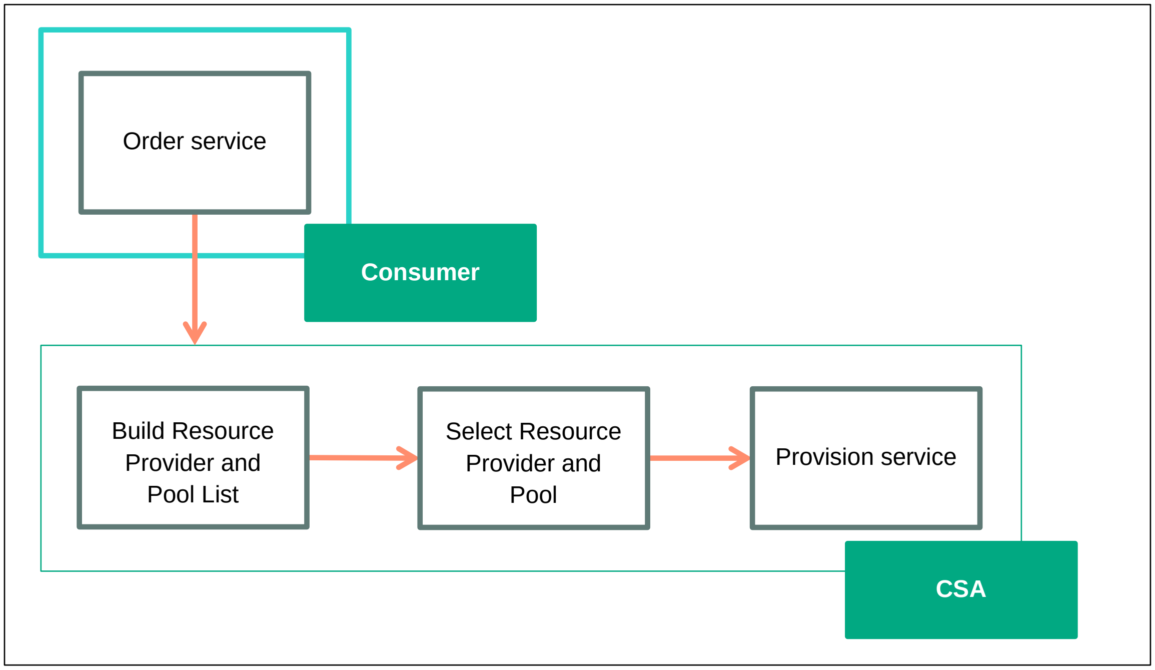
Note: If CSA uses a resource environment associated with a service catalog, it must also select a resource provider associated with that resource environment. To make sure this occurs, the service model *must* specify the Build Resource Provider List and Select Resource Provider actions in the relevant resource binding’s provider-selection actions.

See Use Case: Provider Selection for instructions on how to associate resource environments to providers.

Provider and pool selection

When a consumer orders a service subscription, CSA selects the provider and pool from a list of available resources providers during the reservation phase of the service lifecycle.

Figure 2. Provider and Pool Selection


Order service
Consumer
Build Resource Provider and Pool List
Select Resource Provider and Pool
Provision service
CSA

CSA includes resource provider and pool selection actions by default. These actions select a list of resource providers and pools that are capable of fulfilling the requirements of a resource provisioning request, and then select a provider and pool from the list. The selection can be further refined by filtering the list of providers based on a desired condition before the final resource provider and pool is selected.

The following internal actions are used during provider selection:

- **Build Resource Provider and Pool List** – This action does the following:
  - Builds a list of resource providers and pools.
  - Filters the list that belongs to at least one resource environment that is associated with the corresponding service catalog. This catalog is where service offerings are published that are created from a service design with this action.

    The following criteria are used to build a list of candidate resource providers and pools:
    - Must have the provider availability set to Enabled.
    - Must have the pool availability set to Enabled.
    - The pool must meet the requested demand for the specified resources in the pool.
  - This action also sets the valid list of resource providers and pools to the resulting list.
- **Select Resource Provider and Pool** – This action selects a resource provider and pool from the list built by the previous action. The provider is selected randomly from the list. The selected provider is written to a property on the associated service component. You can specify the name of the property on the action. . If the specified property name does not exist on the associated service component, the property is automatically created on the service component.

Selecting a resource provider and pool from parent

Selects the resource pool and provider that was chosen by a service component's parent service component, as identified by the **Parent Component ID** and **Pool Property Name** properties. The selected resource provider and pool will be available to the resource offering actions in the token RSC_PROVIDER_ID and RSC_POOL_ID. The selected pool id will also be written to a property on the associated service component.

Resource pool selection

A resource pool models the capacity and utilization of resources, which are represented as elements in the service model. Resource pools are optionally configured to represent the resources that a resource provider makes available to CSA.

CSA can model resource supply in resource pools that are associated with each resource provider. For example, you can create resource pools on a VMware vCenter resource provider to represent collections of virtual resources (for example, CPU, memory for a single hypervisor, a cluster, or the entire datacenter). CSA models resource demand using properties on service design components. CSA links the property that expresses a resource demand to a specific resource pool's supply via resource binding actions during service instantiation.

The property on a service design that links to a specific resource pool via a resource binding is known as a *measurable property*. A measurable property has a resource type and a unit of measurement.

CSA allows to you define the following resource types for each pool:

- CPU
- IPv4 address
- IPv6 address
- License
- Memory
- Physical server
- Power
- Storage
- Subnet
- Virtual Server
- VLAN

Resource accounting internal actions

CSA provides the following internal actions to consume or release the resource types defined for the pool:

- **Increase Resource Utilization** – This action is configured during the Reserving transition state. It decrements the Available to CSA resource type in a resource pool by the values of the measurable properties that are configured on a service component.
- **Decrease Resource Utilization** – This action is configured during the Un-reserving transition state and increments the Available to CSA resource type in a resource pool by the values of the measurable properties that are configured on a resource binding.

Sequenced design use cases

A service design is composed of components and resource offerings to make up an orderable service. Resource offerings that are bound to components will need a provider instance when the service is fulfilled. If multiple provider instances are available during fulfillment, then CSA can choose a provider instance or delegate the choice to an external system. Such a choice can be modeled by the designer in the service design.

Similarly, each provider instance can have multiple resource pools. The designer can either configure the provider to choose a resource pool or delegate to an external system to make the choice.

Several resource management use cases can be configured in CSA for an orderable service:

- Provider selection (default behavior)
- Custom provider selection during fulfillment
- Custom provider and pool, with pool selected during fulfillment
- Resource quota
- Resource synchronization of a pool

Note: These use cases apply only to sequenced designs.

Use case: Provider selection

Let us assume the following scenario: we want to design a service to provision a simple compute server using HPE Matrix Operating Environment (Matrix OE). We have access to three resource providers for this purpose.

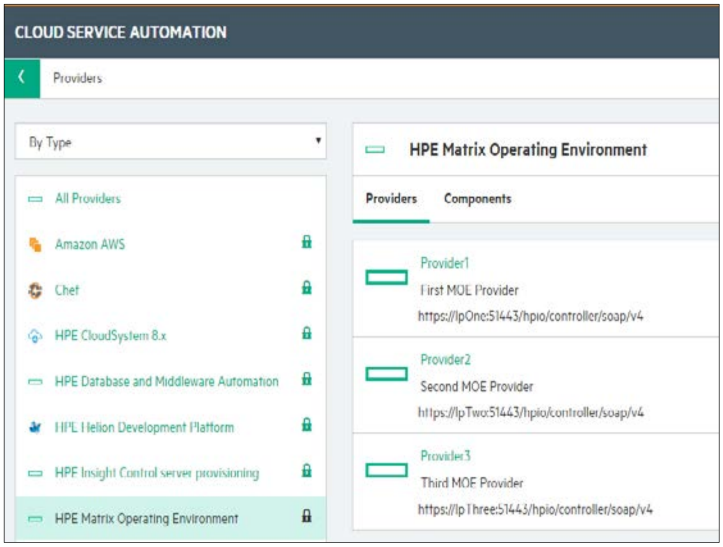
Using this scenario, we will walk through a step by step example. The purpose of this example is to help you understand how to configure provider selection actions and to show you how a service subscription results in the selection of a provider. This example is not a best practice document of how to design a service, which is beyond the scope of this document.

These steps will not illustrate basic CSA operations such as adding a resource provider or publishing a service offering. These basic operations are documented in the *Cloud Service Management Console Help*.

Steps to configure the provider selection

1. Download the content from HPLN: https://hpln.hpe.com/contentoffering/resource-provider-selection3
2. Use the Capsule Installer to Import the content into the CSA instant.
3. Add all three Matrix OE providers to the list of available providers, naming them Provider1, Provider2, and Provider3.

Figure 3. Matrix Resource Providers


CLOUD SERVICE AUTOMATION
Providers
By Type
All Providers
Amazon AWS
Chef
HPE CloudSystem 8.x
HPE Database and Middleware Automation
HPE Helion Development Platform
HPE Insight Control server provisioning
HPE Matrix Operating Environment
Providers
Components

HPE Matrix Operating Environment

Provider1
First MOE Provider
https://IpOne:51443/hpio/controller/soap/v4

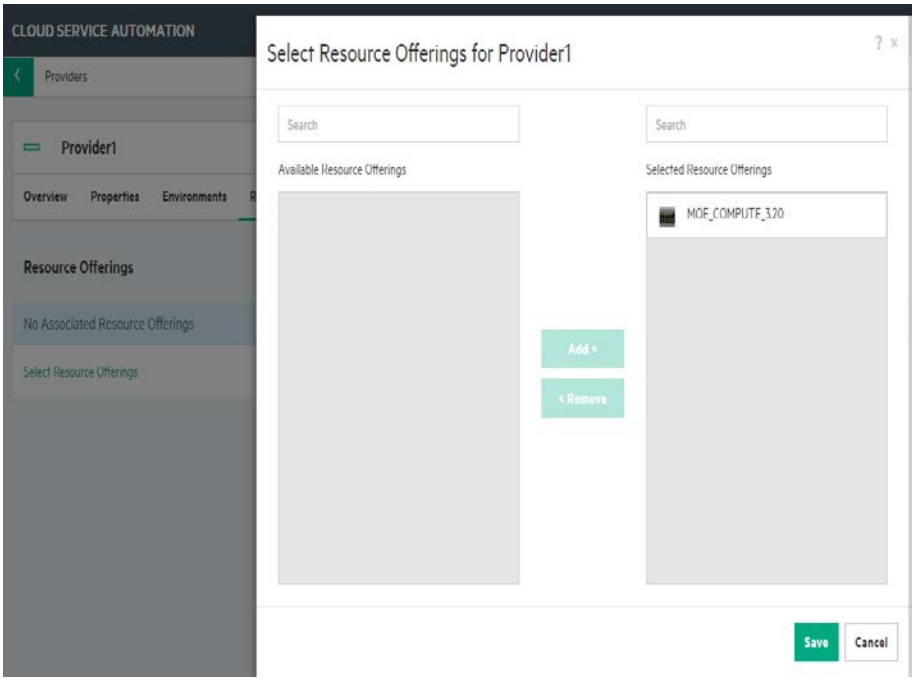
Provider2
Second MOE Provider
https://IpTwo:51443/hpio/controller/soap/v4

Provider3
Third MOE Provider
https://IpThree:51443/hpio/controller/soap/v4

4. Associate resource offerings to resource providers that are capable of offering the resource.

   Because we have three Matrix OE providers that are capable of provisioning a simple compute server, let us associate these three providers with the MOE_COMPUTE resource offering, which is available as supplied content. This is done for each provider. In Figure 4, the MOE_COMPUTE offering is being associated with Provider1.

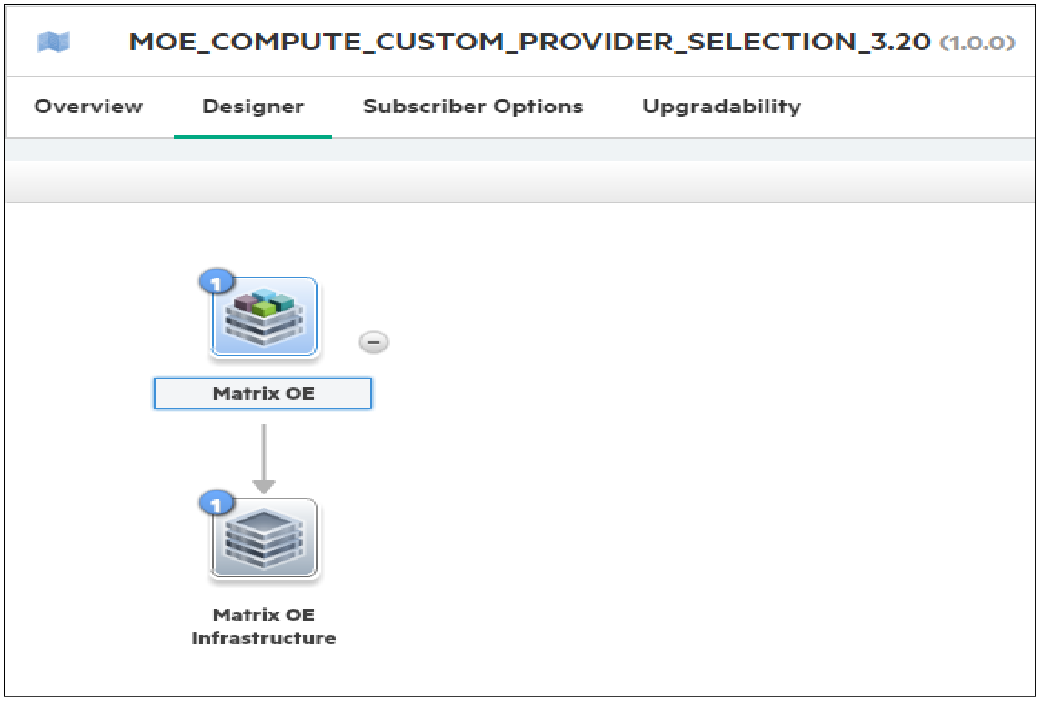
   Figure 4. Provider1 Association with the Matrix OE Compute Resource Offering


CLOUD SERVICE AUTOMATION
Providers
Provider1
Overview
Properties
Environments
Resource Offerings
No Associated Resource Offerings
Select Resource Offerings
Select Resource Offerings for Provider1
? ×
Search
Search
Available Resource Offerings
Selected Resource Offerings
MOE_COMPUTE_3.20
Add ›
‹ Remove
Save
Cancel

5. View the service to provision a simple compute server in the Designer.

   A service that provisions a simple compute server using Matrix OE is included in the CSA installation, and can be viewed in the designer, as shown in Figure 5. Note that this design contains a service composite component (Matrix OE) and an infrastructure service component (MOE Infrastructure) as its child.

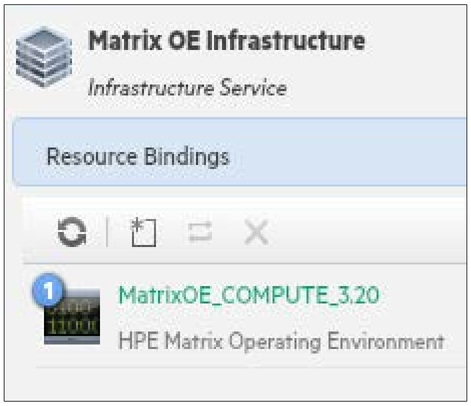
   Figure 5. Provided Matrix OE Service Design



6. Associate the service component to the resource offering.

   As stated in step 3, the MOE_COMPUTE resource offering is available as supplied content with CSA. In this step, we associate this resource offering to the infrastructure service (MOE Infrastructure) component through a resource binding. A resource binding is a link between a resource offering and a service component. It ensures that the resource offering will be provisioned when the service component is deployed for a subscription.

   Figure 6. Resource Binding to the MOE_COMPUTE Resource Offering



Matrix OE Infrastructure

Infrastructure Service

Resource Bindings

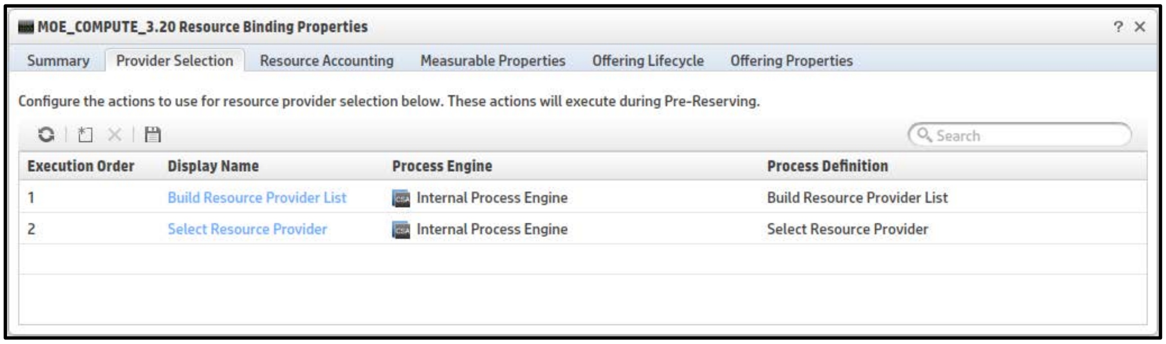
1 MatrixOE_COMPUTE_3.20

HPE Matrix Operating Environment

7. Create provider selection actions for the resource bindings.

   The next step is to create provider selection actions (discussed in the Provider selection section) for the resource binding. When there is a subscription to the service, CSA creates a service instance which moves through the CSA lifecycle phases. During that time, provider selection actions will be executed during the Reserving – Pre-Transition phase and will result in selection of a resource provider to provide the resource offering, the provisioning of the simple compute server.

   Figure 7. Provider Selection Actions in the Matrix OE Resource Binding



Configure the actions to use for resource provider selection below. These actions will execute during Pre-Reserving.

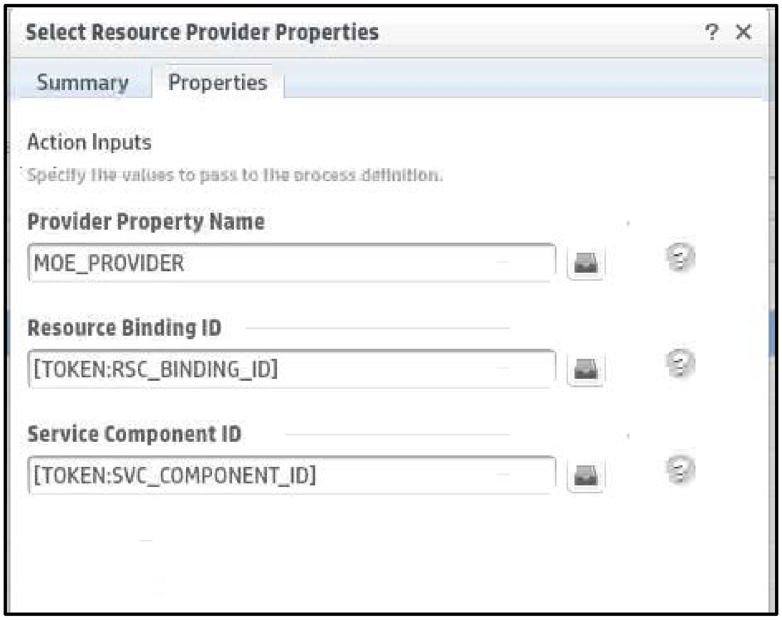
Execution Order	Display Name	Process Engine	Process Definition
1	Build Resource Provider List	Internal Process Engine	Build Resource Provider List
2	Select Resource Provider	Internal Process Engine	Select Resource Provider

8. Specify the properties of the internal actions.

   The Select Resource Provider action has a property named Provider Property Name. With this property, you can specify the name of the property in which the selected resource provider ID will be set as a property of the infrastructure service component. For example, if you specify MOE_PROVIDER as the value for this property, then the selected resource provider will be set as a property on the service component as follows:

   Provider=ID of Selected Provider.

Figure 8. Provider Property Name



Select Resource Provider Properties

Action Inputs

Specify the values to pass to the process definition.

Provider Property Name: MOE_PROVIDER

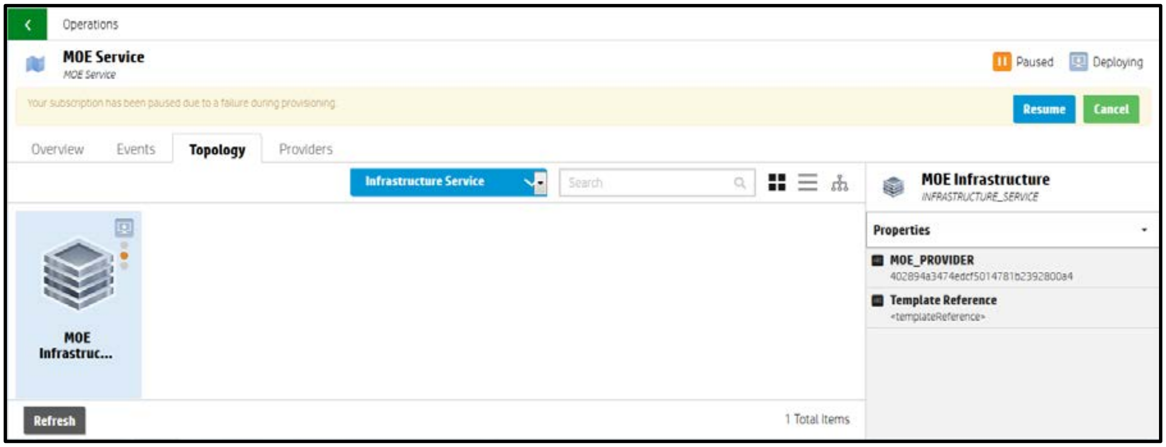
Resource Binding ID: [TOKEN:RSC_BINDING_ID]

Service Component ID: [TOKEN:SVC_COMPONENT_ID]

After the above steps are performed, the service is ready to be associated with a service offering and published to a catalog. After a consumer subscribes to this service, CSA creates a service instance that moves through the CSA lifecycle phases. When the provider selection actions are executed during the Reserving – Pre-Transition lifecycle phase, then the following sequence of operations takes place:

- The Build Resource Provider List action runs, which does the following:
  a. Gets the list of resource providers associated with the resource offering corresponding to the resource binding. This returns Provider1, Provider2, and Provider3.
  b. Gets the list of resource environments associated with the service catalog (which contains the service offering based on our Matrix OE service design).
  c. Filters the list returned above to include only those providers which are associated with the resource environments of the service catalog. For example, assume that the catalog is associated with a resource environment that contains two resource providers – Provider1 and Provider2. From the list returned above, Provider3 will not be included. So the resulting list will include just Provider1 and Provider2.
  d. Checks the list of providers returned above to see if each provider is enabled, and includes only the list of enabled providers. In our example, all providers are enabled, so both Provider1 and Provider2 will be returned.
  e. Finally sets the list of candidate providers to the list returned above. It also sets the list of valid providers to the same list. In our example, this will be set to Provider1 and Provider2.
- The Select Resource Provider action runs, which does the following:
  a. Gets the list of valid providers built by the Build Resource Provider List action. This includes Provider1 and Provider2.
  b. Selects a resource provider based on random selection from this list. Let us assume the selected provider is Provider1.
  c. Sets the provider selected above as the selected resource provider of the resource binding. In our example, this is Provider1.
  d. Sets the ID of the selected provider as a property of the service component. The name of this property is the user entered value for Provider Property Name when this action was added to the resource binding (See Figure 8). In our example, this is set as MOE_PROVIDER= 402894a3474edcf5014781b2392800a4 (See Figure 9).

Figure 9. Provider Property of Service Component



MOE Service
MOE Service

Your subscription has been paused due to a failure during provisioning

1 Total Items

MOE Infrastructure
INFRASTRUCTURE_SERVICE

MOE_PROVIDER
402894a3474edcf5014781b2392800a4

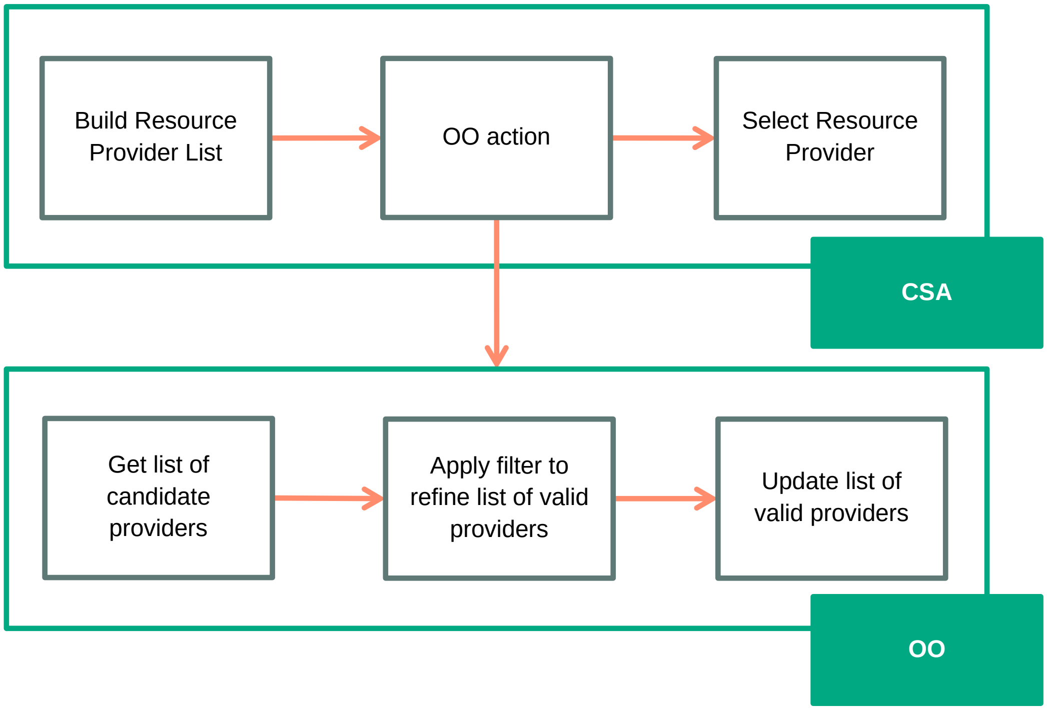
Template Reference
<templateReference>

Use case: Custom provider selection

Using custom provider selection, the default provider selection functionality described above can be refined to further filter the list of resource providers based on some condition decided by the user.

This requires that the initial list of resource providers built by the Build Resource Provider List action be further refined before the Select Resource Provider action chooses the resource provider. This can be achieved using an OO flow which performs the filtering functionality. After importing this flow into CSA, the corresponding CSA action can be inserted between the Build Provider List and Select Resource Provider actions. Figure 10 illustrates this concept.

Figure 10. Implementing Custom Provider Selection


Build Resource Provider List
OO action
Select Resource Provider
CSA
Get list of candidate providers
Apply filter to refine list of valid providers
Update list of valid providers
OO

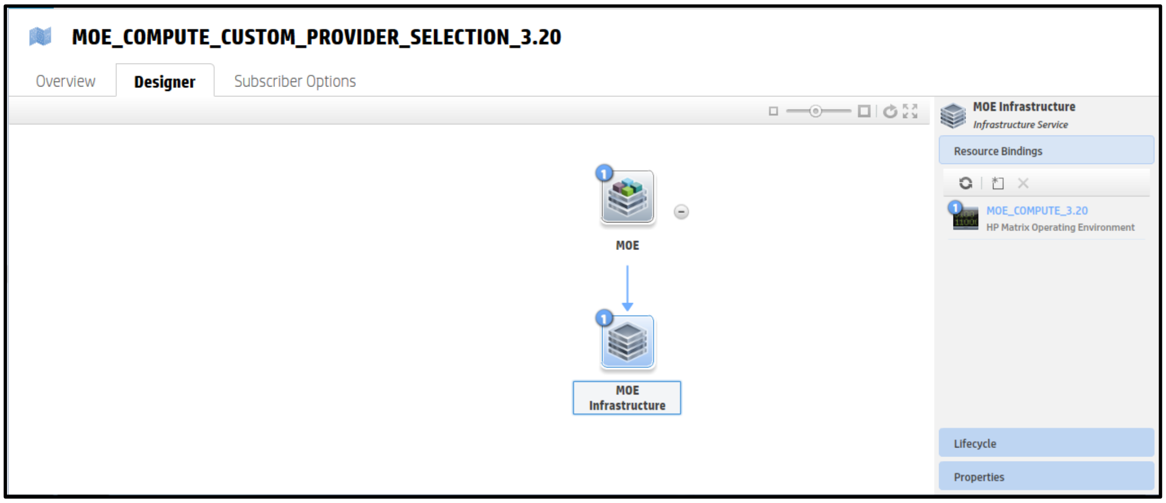
As shown in Figure 10, the OO action performs three major functions:

- Gets the list of candidate providers built by the Build Resource Provider List action.
- Refines the list with the filtering condition decided by the user.
- Updates the list of valid providers in CSA. This updated list is used as the basis for selection by the Select Resource Provider action.

To enable OO to interact with CSA, CSA exposes appropriate REST APIs. To get more information on REST APIs, refer to the *CSA API Guide*. The relevant REST APIs to implement custom provider selection are explained in the example in the next section.

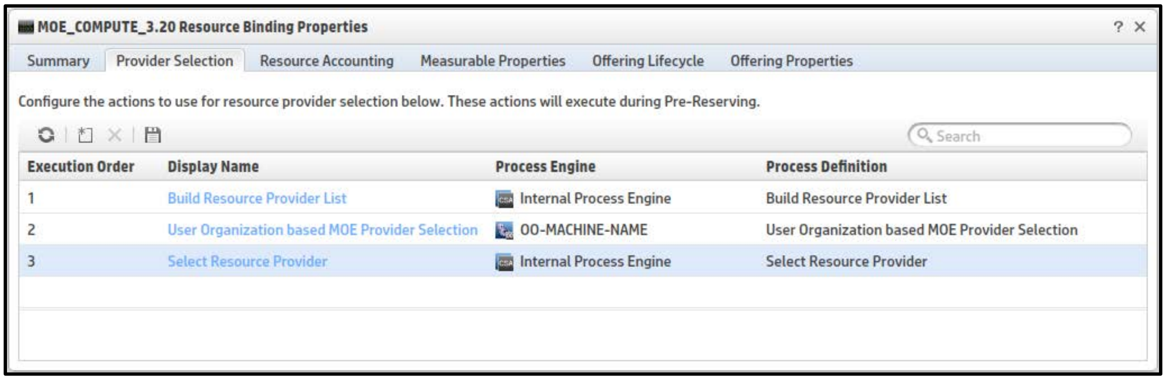
For this example, we will use the MOE_COMPUTE_CUSTOM_PROVIDER_SELECTION service design, which is part of the sample content shipped with CSA.

Figure 11. Sample Matrix OE Service Design for Custom Provider Selection



From Figure 11, we can see that the MOE Infrastructure component has a resource binding to the MOE_COMPUTE resource offering. Let us look at the provider selection actions configured on this resource binding.

Figure 12. Provider Selection Actions



Configure the actions to use for resource provider selection below. These actions will execute during Pre-Reserving.

Execution Order	Display Name	Process Engine	Process Definition
1	Build Resource Provider List	Internal Process Engine	Build Resource Provider List
2	User Organization based MOE Provider Selection	OO-MACHINE-NAME	User Organization based MOE Provider Selection
3	Select Resource Provider	Internal Process Engine	Select Resource Provider

Compared to our previous example, this resource binding has an additional action between the Build Resource Provider List and Select Resource Provider actions. This is an OO flow that refines the list of providers further before the final resource provider is selected.

This action is available because the corresponding OO flow has been imported into CSA. To implement custom provider selection, we need to define our own flow and import it into CSA. For details on how to import flows into CSA, see the *CSA Configuration Guide*.

At a broad level, the flow corresponding to this action performs three major functions:

1. Gets the list of candidate providers using the CSA REST API. See Appendix A: REST APIs for custom provider selection for details.
2. Refines the list of providers based on some condition. This condition depends on the specific use case of the customer. For instance, taking our previous example, we may want to provide services to users from two organizations – Organization A, and B. We want Provider1 and Provider2 to provide resources for users from Organization A and Provider3 to provide resources to users from Organization B. To implement this condition, we can define a property on all resource providers to indicate the acceptable organization (Organization=A for Provider1 and Provider2, Organization=B for Provider3). In the OO flow, the list returned in step 1 should be filtered to include only those providers whose Organization property value includes the requesting user's organization. If the requesting user belongs to Organization A, then only Provider1 and Provider2 will be set as the valid providers. At the end of the refinement process, a subset of the providers will be valid choices.
3. Updates the list of valid providers with the list returned in step 2 using the CSA REST API. See Update the list of valid providers.

Details of how to create an OO flow are beyond the scope of this document. To learn how to create flows, see the *HPE Operations Orchestration Studio Authoring Guide*. A basic overview of the flow (User Organization based MOE Provider Selection) is presented in Appendix A: REST APIs for custom provider selection. This flow is shipped as sample content with CSA.

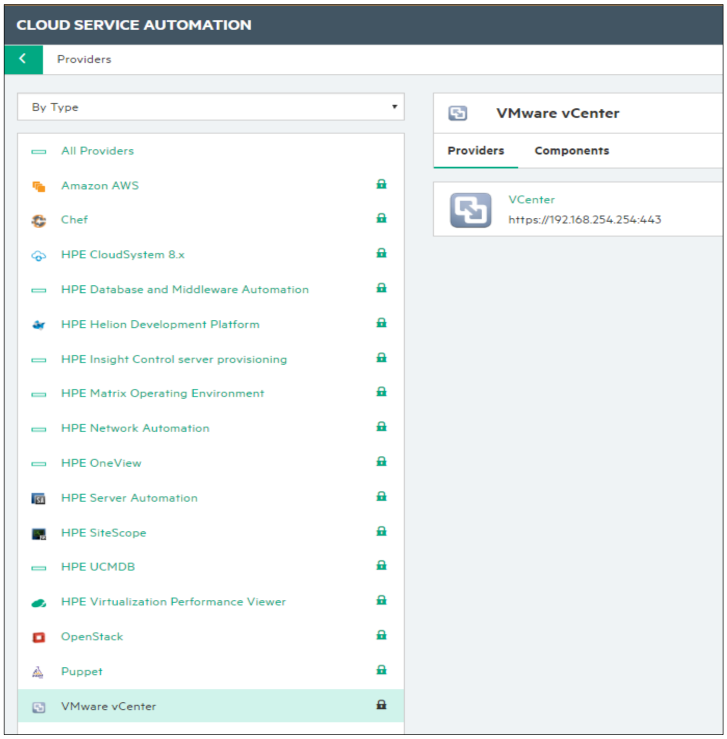
Use case: Provider and pool selection

Data store is an essential resource capacity that is consumed by various applications in a data center. Consider a VMware vCenter provider having three data stores in a data center. Subscription request fulfillment requires selecting a data store having enough disk space to provision the server instance. Each data store can be configured as a provider pool on the VCenter resource provider. During provider selection, appropriate data stores can be selected by comparing the disk size request from the customer request against the provider's data store disk space.

Steps to configure provider selection

1. Download the content from HPLN : https://hpln.hpe.com/contentoffering/resource-provider-selection3
2. Import the content using Capsule Installer into CSA instance
3. Add a vCenter resource provider.

   Figure 13. vCenter Resource Provider

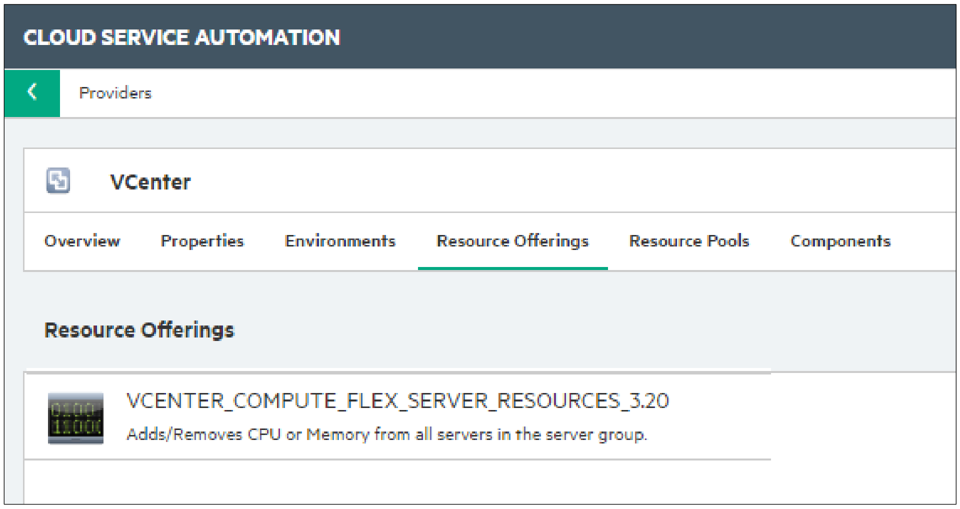


VCenter
https://192.168.254.254:443

4. Associate resource offerings to resource providers that are capable of offering the resource.

   For this example, we will associate the VMware vCenter provider with the resource offering named VCENTER_FLEX_COMPUTE_SERVER_RESOURCES.

Figure 14. vCenter Flex Resource Offering



Resource Offerings

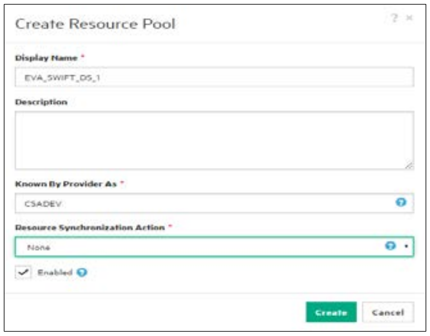
VCENTER_COMPUTE_FLEX_SERVER_RESOURCES_3.20

Adds/Removes CPU or Memory from all servers in the server group.

5. Create resource pools.

Create data store pools on CSA so that they have display names that are the same as the name on the VMware vCenter provider. Figure 15 shows that there are three data store pools for the provider, and that they have been given the same display names in CSA.

Figure 15. Datastore Pools



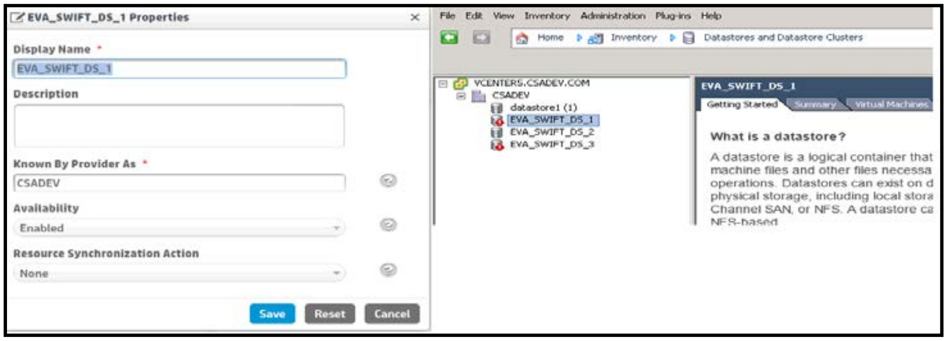
Display Name *: EVA_SWIFT_DS_1

Description:

Known By Provider As *: CSADEV

Resource Synchronization Action *: None

☑ Enabled



Display Name *: EVA_SWIFT_DS_1

Description:

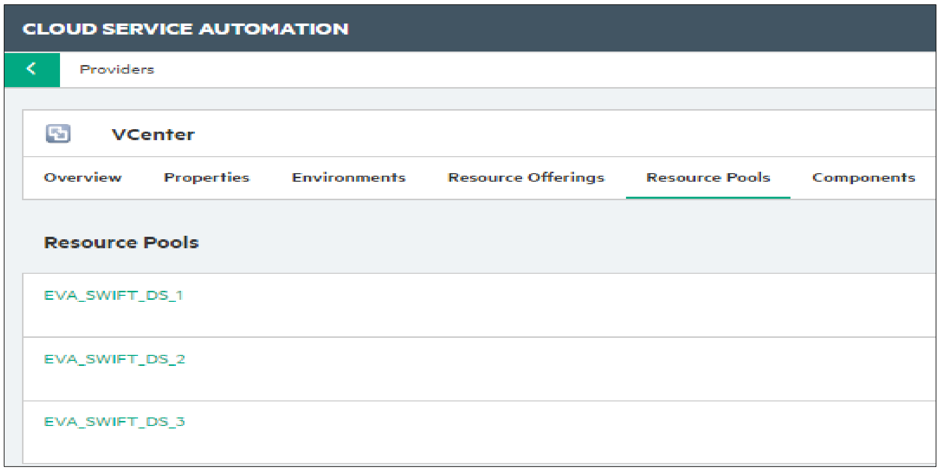
Known By Provider As *: CSADEV

Availability: Enabled

Resource Synchronization Action: None

What is a datastore?

A datastore is a logical container that machine files and other files necessa operations. Datastores can exist on d physical storage, including local stora Channel SAN, or NFS. A datastore ca NFS-based



Resource Pools

EVA_SWIFT_DS_1

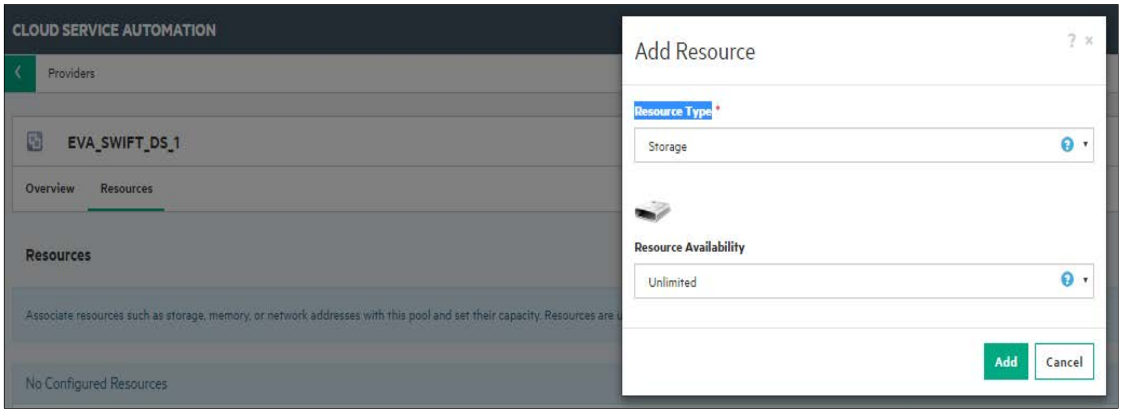
EVA_SWIFT_DS_2

EVA_SWIFT_DS_3

6. Add the resource capacity and availability for each resource pool.

   Add storage capacity for the data store resource pool. Select Unlimited for the resource availability.

   Figure 16. Resource Capacity of a Pool



Resources

Associate resources such as storage, memory, or network addresses with this pool and set their capacity. Resources are u

No Configured Resources

Add Resource

Resource Type *: Storage

Resource Availability: Unlimited

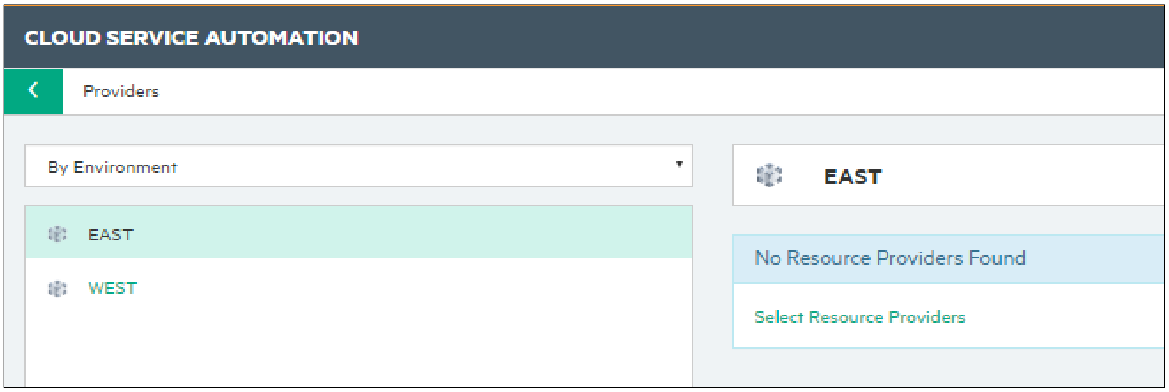
7. Add the providers to the resource environment.

Note: This is an optional step during provider configuration.

Providers can be grouped together by *resource environment*. One or more resource environments can be linked to a service catalog to restrict provider selection at subscription time. The steps to do this are indicated in the following example:

   a. Add two regions, East and West, by first selecting **By Environment** in the upper-left drop-down field and clicking on the gear icon to add the regions.

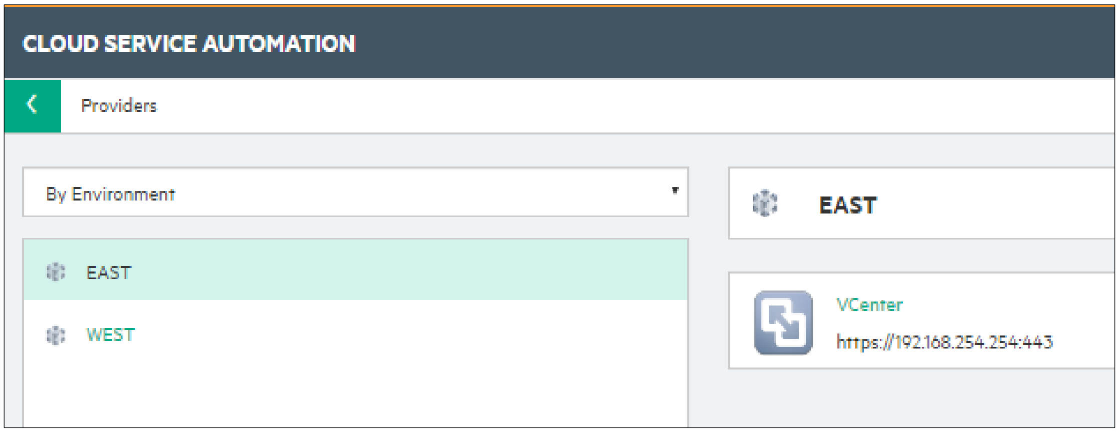
      Figure 17. East and West Regions Added



No Resource Providers Found

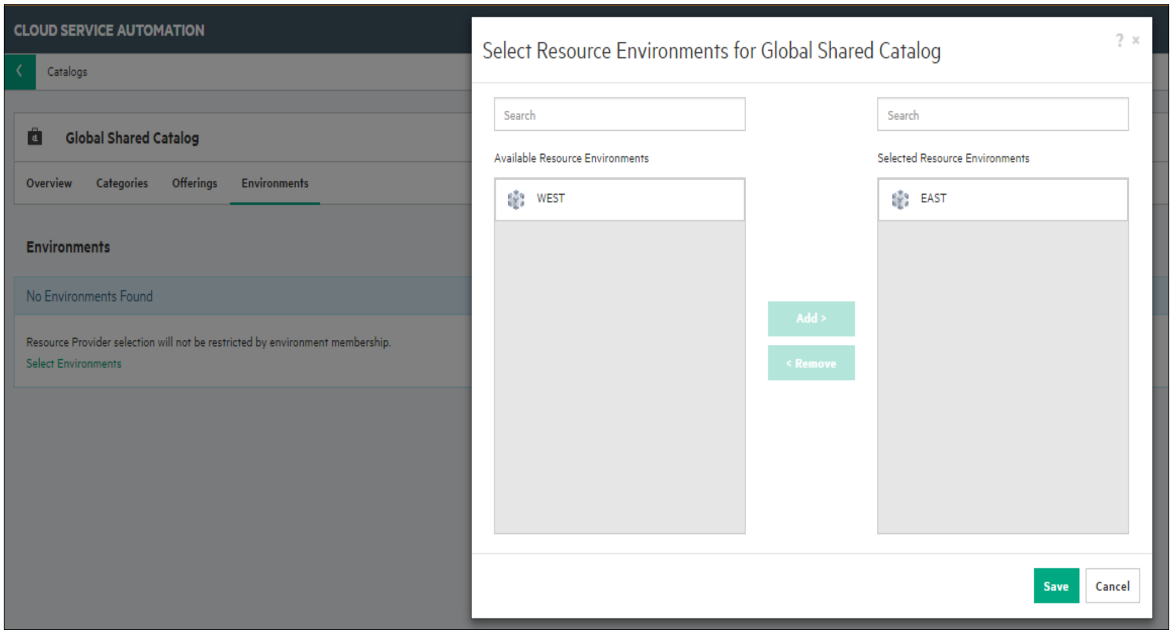
b. Associate the necessary providers into each resource environment.

Figure 18. vCenter Provider Associated with the East Region



c. Add the resource environments onto the desired catalogs in the Catalog area of the CSA interface. In this example, we will add the East resource environment into the Global catalog, so subscriptions requested from the Global catalog will always choose providers from the East resource environment during the provider selection stage of subscription fulfillment.

Figure 19. Adding a Resource Environment to a Catalog



No Environments Found

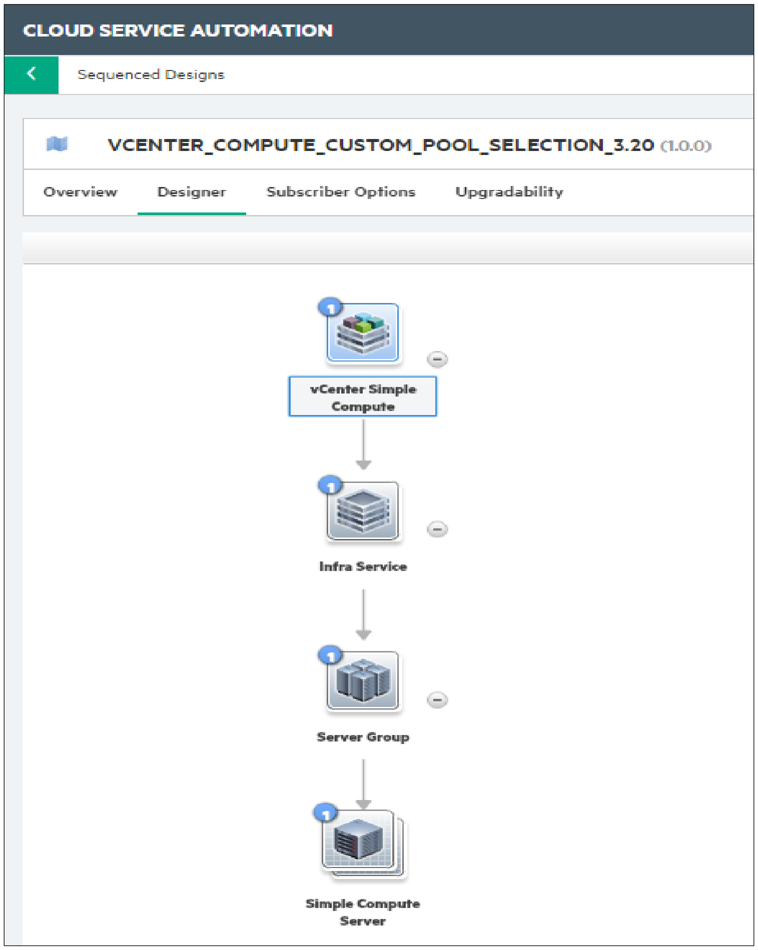
Resource Provider selection will not be restricted by environment membership.

8. Create a sequenced service design.

For this example, create a sequenced service design having the following components, with the hierarchy as indicated in the following list and shown in Figure 20:

- Service Composite
- Infrastructure Service
- Server Group
- Server

Figure 20. Service Design


CLOUD SERVICE AUTOMATION

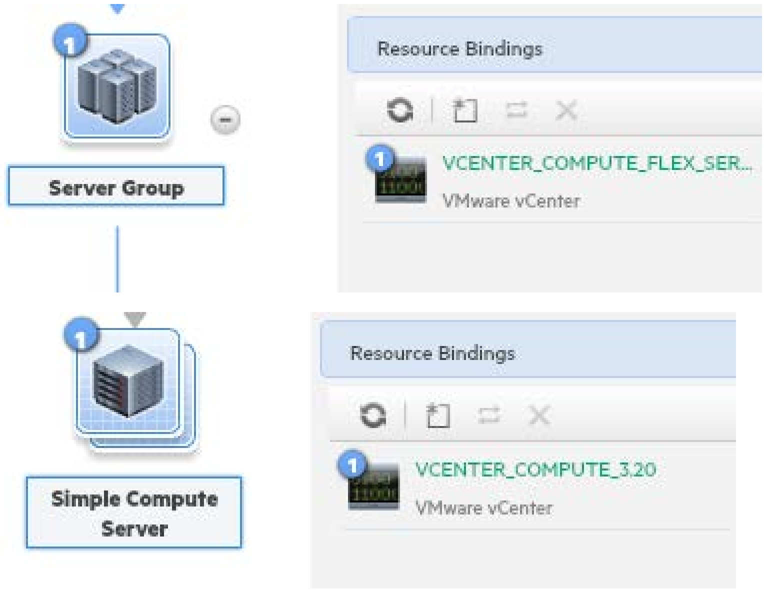
Sequenced Designs

VCENTER_COMPUTE_CUSTOM_POOL_SELECTION_3.20 (1.0.0)

9. Add the resource offerings to the components.

Resource offerings are associated with components using resource bindings. In this example, add the VCENTER_FLEX_SERVER_RESOURCES binding to the Server Group component, and the VCENTER_COMPUTE binding to the Server component.

Figure 21. Resource Offering on Components



Resource Bindings

VCENTER_COMPUTE_FLEX_SER...
VMware vCenter

Resource Bindings

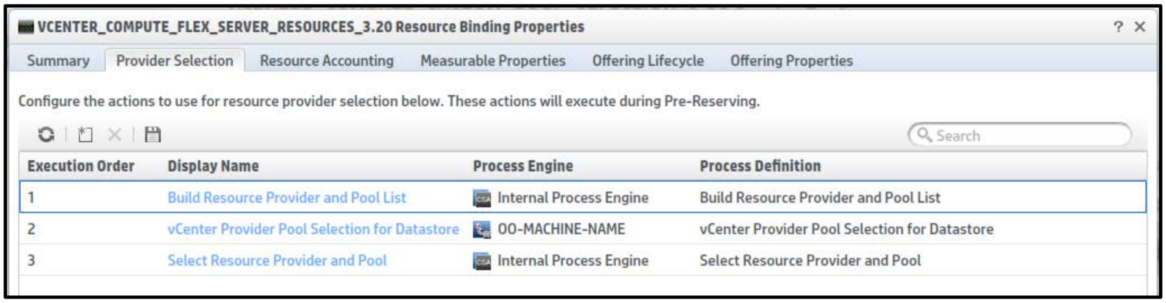
VCENTER_COMPUTE_3.20
VMware vCenter

10. Create Provider Selection actions for the resource bindings on the Server Group component.

    Add the following actions to the Server Group component's binding (VCENTER_FLEX_SERVER_RESOURCES) in the following order:

    - Build Resource Provider and Pool List (Internal Process Engine)
    - vCenter Provider Pool Selection for Datastore (OO Flow)
    - Select Resource Provider and Pool (Internal Process Engine)

    Figure 22. Provider and Pool Actions on the Resource Binding for the Server Group



VCENTER_COMPUTE_FLEX_SERVER_RESOURCES_3.20 Resource Binding Properties

Configure the actions to use for resource provider selection below. These actions will execute during Pre-Reserving.

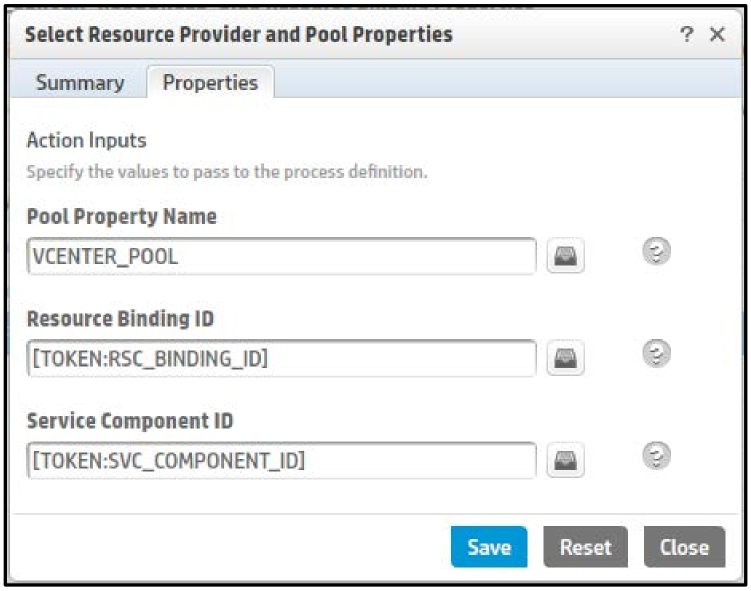
Execution Order	Display Name	Process Engine	Process Definition
1	Build Resource Provider and Pool List	Internal Process Engine	Build Resource Provider and Pool List
2	vCenter Provider Pool Selection for Datastore	OO-MACHINE-NAME	vCenter Provider Pool Selection for Datastore
3	Select Resource Provider and Pool	Internal Process Engine	Select Resource Provider and Pool

11. Specify the property of the Select Resource Provider and Pool internal action.

    The Select Resource Provider and Pool action has a property called Pool Property Name. With this property the user can specify the name of the property in which the selected resource provider ID will be set as a property of the service component. For example, if the user specifies VCENTER_POOL as the value for this property, then the selected provider pool Id will be set as a property on the service component as follows:

    Provider=ID of Selected Provider Pool.

    Figure 23. Provider Property Name



Select Resource Provider and Pool Properties

Action Inputs

Specify the values to pass to the process definition.

Pool Property Name: VCENTER_POOL

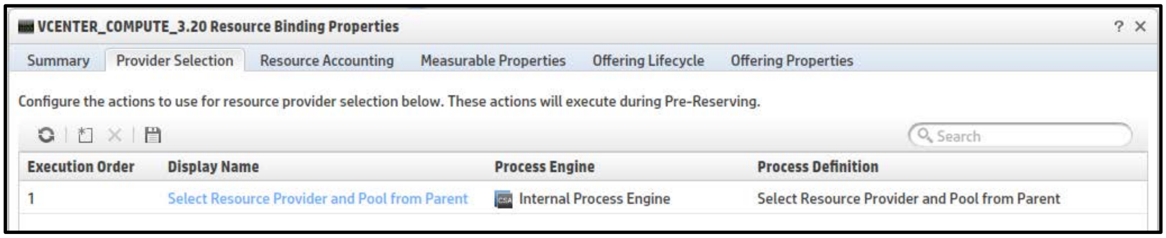
Resource Binding ID: [TOKEN:RSC_BINDING_ID]

Service Component ID: [TOKEN:SVC_COMPONENT_ID]

12. Create provider selection actions for the resource bindings on the server component.

    Add the Select Resource Provider and Pool from Parent internal action on the server component's binding (VCENTER_COMPUTE).

    Figure 24. Provider Selection Action on Resource Binding



VCENTER_COMPUTE_3.20 Resource Binding Properties

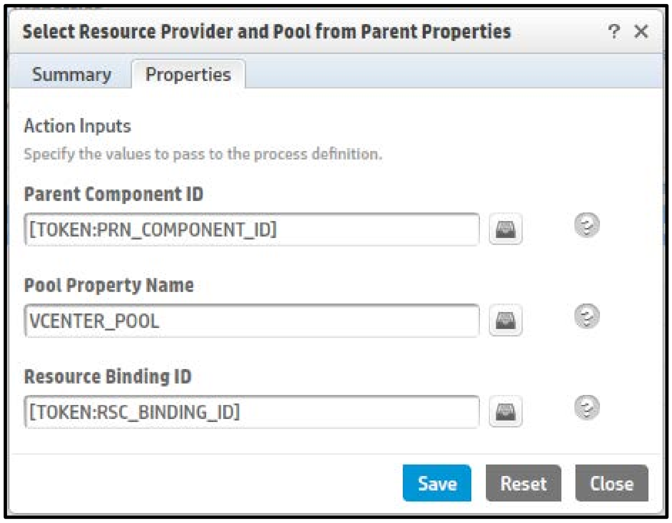
Configure the actions to use for resource provider selection below. These actions will execute during Pre-Reserving.

Execution Order	Display Name	Process Engine	Process Definition
1	Select Resource Provider and Pool from Parent	Internal Process Engine	Select Resource Provider and Pool from Parent

13. Specify the property of the internal action.

    The Select Resource Provider and Pool from Parent action has a property called Pool Property Name. Set the value of this property to VCENTER_POOL. The action will retrieve the Provider Pool Id populated on the property VCENTER_POOL from the parent server group component and store it on the server component.

    Figure 25. Provider Property Name



Select Resource Provider and Pool from Parent Properties

Action Inputs

Specify the values to pass to the process definition.

Parent Component ID: [TOKEN:PRN_COMPONENT_ID]

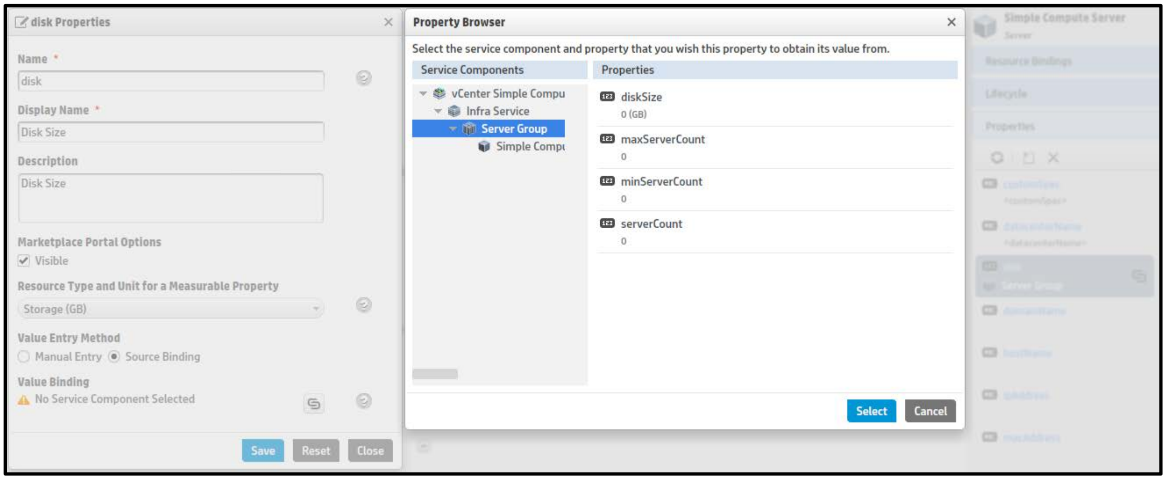
Pool Property Name: VCENTER_POOL

Resource Binding ID: [TOKEN:RSC_BINDING_ID]

14. Create the following component properties on the Server Group component:
    - An Integer property named "diskSize"
    - A String property named "templateReference"
15. Source bind the server component properties to the server group component, as follows:
    - Source bind the disk property on the server component to the diskSize property on the server group component (Figure 26).
    - Source bind the templateReference property on the server component to templateReference property on the server group component (Figure 27)

    Note: Source bind is a link that is created between a child component property to a parent component property

Figure 26. Source Binding the diskSize Property



disk Properties

Name *
disk

Display Name *
Disk Size

Description
Disk Size

Marketplace Portal Options
☑ Visible

Resource Type and Unit for a Measurable Property
Storage (GB)

Value Entry Method
○ Manual Entry ● Source Binding

Value Binding
No Service Component Selected

Save Reset Close

Property Browser

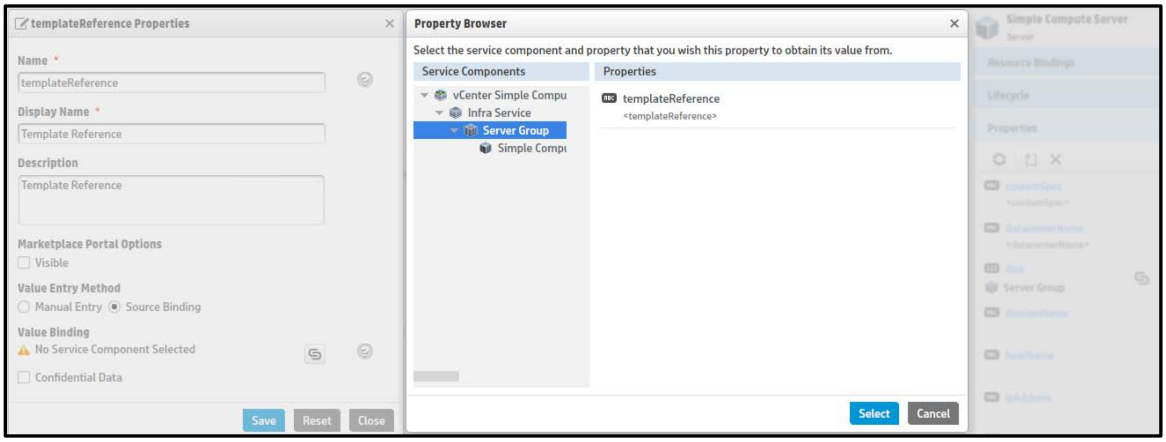
Select the service component and property that you wish this property to obtain its value from.

Service Components
- vCenter Simple Compu
  - Infra Service
    - Server Group
      - Simple Compu

Properties
- diskSize
  0 (GB)
- maxServerCount
  0
- minServerCount
  0
- serverCount
  0

Select Cancel

Figure 27. Source Binding the templateReference Property



templateReference Properties

Name *
templateReference

Display Name *
Template Reference

Description
Template Reference

Marketplace Portal Options
□ Visible

Value Entry Method
○ Manual Entry ● Source Binding

Value Binding
No Service Component Selected

□ Confidential Data

Save Reset Close

Property Browser

Select the service component and property that you wish this property to obtain its value from.

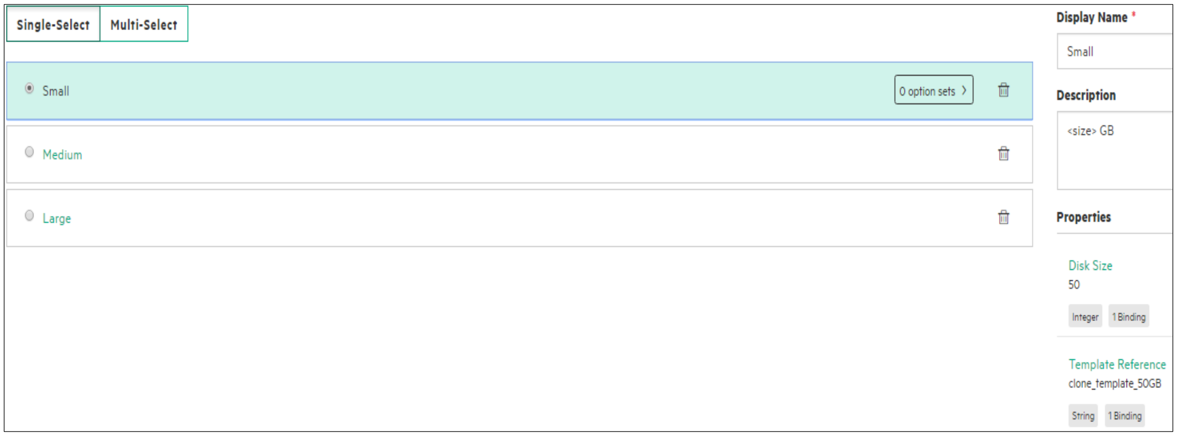
Service Components
- vCenter Simple Compu
  - Infra Service
    - Server Group
      - Simple Compu

Properties
- templateReference
  <templateReference>

Select Cancel

16. Create the necessary subscriber options, as follows:
    a. Create an option set having options for various disk sizes (Small and Medium, for example, in Figure 28).
    b. Create an integer property named DiskSize on each option having values corresponding to the storage sizes.
    c. Create a string property named templateReference on each option, with the value being a clone template name residing on the vCenter provider. The clone template should provision the instance for the requested disk size.

    Figure 28. Subscriber Options



Single-Select | Multi-Select

- ● Small — 0 option sets
- ○ Medium
- ○ Large

Display Name *
Small

Description
<size> GB

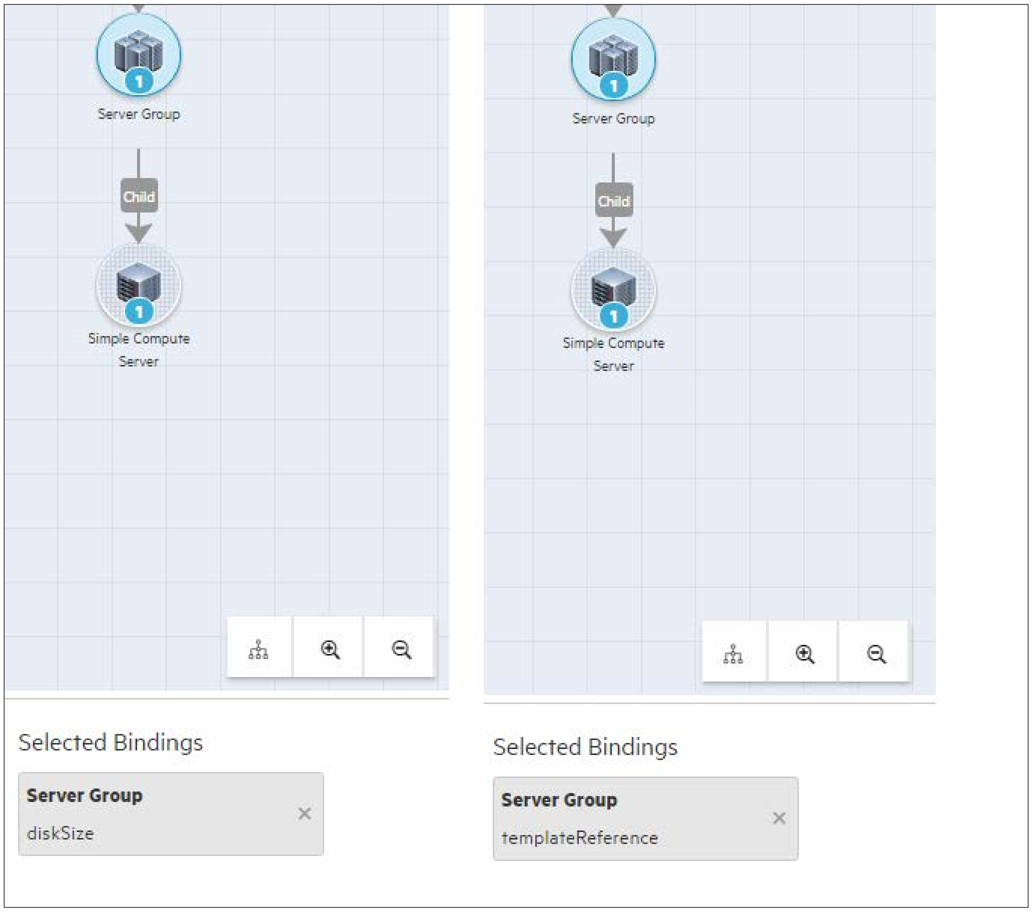
Properties

Disk Size
50
Integer | 1 Binding

Template Reference
clone_template_50GB
String | 1 Binding

    d. Create a target binding between the DiskSize property on each option to the diskSize property on the server group.
    e. Create target binding between the templateReference property on each option to the templateReference property on the server group.

Figure 29. Target Binding the Option Properties to Server Group Properties



Selected Bindings

Server Group
diskSize

Selected Bindings

Server Group
templateReference

17. Publish the design.

    After you have completed the steps outlined above, the service is ready to be associated with a service offering and published to a catalog. After a consumer subscribes to the published service, CSA creates a service instance that moves through the CSA lifecycle phases. When the provider selection actions are executed during the Reserving – Pre-Transition lifecycle phase, the following sequence of operations occurs:

- **Build Resource Provider and Pool List** is an internal action that gets the list of resource providers associated with the resource offering corresponding to the resource binding, and CSA stores the vCenter provider and the list of pools associated with the provider. The list stored in CSA for the binding is called the Candidate Provider and Pool list.
- **vCenter Provider Pool Selection for Datastore** is an OO flow that performs the following steps:
  - Retrieves the list of data store names and corresponding storage space from the VMware vCenter provider.
  - Retrieves the Candidate Provider and Pool list stored in the previous action using the CSA REST API. See Appendix C.
  - Retrieves the disk size property value defined on the Server Group component.
  - Verifies the pool names for each of the above lists against the candidate pool list.
  - Filters the candidate pool list which satisfies the storage space against the disk size.
  - Updates the filtered candidate list as a valid pool list onto CSA using the CSA REST API. See Appendix C.
- **Select Resource Provider and Pool** is an internal action that performs following steps:
  - Retrieves the list of valid provider and pools built by the previous action.
  - Selects a random pool Id and saves it as the VCENTER_POOL property on the Server Group component.

Note: See OO documentation for more information on creating OO flows.

Use case: Managing supply and demand

You can manage supply and demand using resource pools.

Resource administrators manage resource quota to take care of supply management. The following summarizes supply management in CSA.

- To configure a resource supply, an administrator must identify the capacity of each resource that can be allocated for a provider and pool.
- Multiple pools can be configured per provider.
- Multiple resources per pool are supported. For example, a pool can have CPU, memory, and storage resources configured.
- You can associate an environment with the provider.

Resource demand management

The following summarizes demand management in CSA.

- Resource demand is expressed using the Measurable properties, which are described as properties on a Service Component. Measurable properties have a resource type and a unit of measurement.
- A resource or multiple resources (CPU, storage, etc.) in demand will be presented to provider and resource pools. The provider and the pool which satisfy the resource demand will be selected for utilization.
- If resource demand is available external to CSA during provider selection, then one can externalize the provider and pool selection. One must develop an OO flow that will determine the resource demand and select the appropriate provider and pool. See Use case: Provider and pool selection for details.
- When the demand criteria is satisfied, the accounting actions will utilize the resources and decrement the allocated capacity of resources from the resource pool. These actions are defined in the Reserve and Un-Reserve transitions phases.

Use case: Resource pool

CSA administrators configure resource pools to utilize the resources of the provider. The following uses cases can be derived while configuring providers and pools.

Case 1: Administrators must know the available capacity for each resource that must be configured in the CSA resource pool. During provider and pool selection, these resources will be consumed accordingly.

Case 2: Administrators cannot determine available resource capacity to configure on the resource pool. In such cases they can still configure the pool but externalize the “provider and pool selection” by determining the resources available on the actual provider.

As an example for case 1 above, let us assume that we want to design a service that will provision a simple compute server using the provider VMWARE VCENTER . The service should consume resources from the pool while provisioning and release resources back to the pool during tear down of the service.

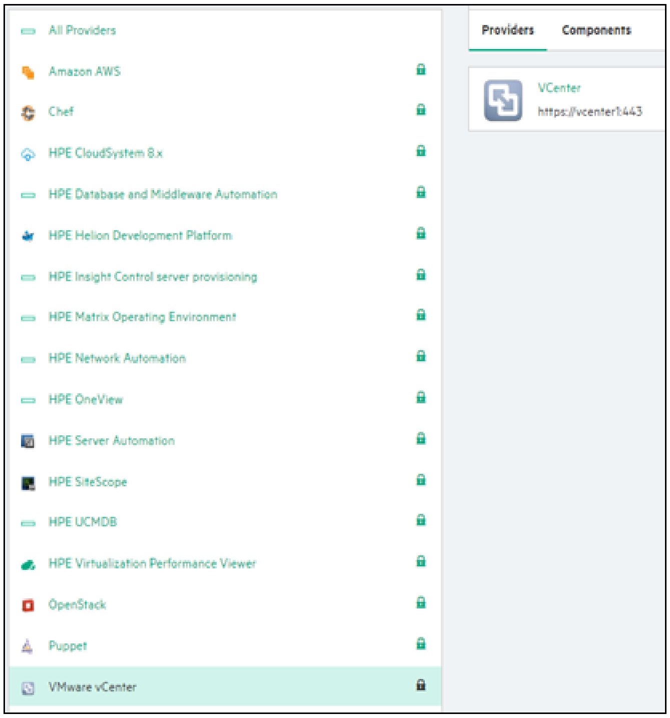
The purpose of this example is to help the reader understand how to configure resource pools, configure resource types for each pool, and configure measurable properties and internal actions on resource bindings.

The steps will not illustrate basic operations, such as adding a resource provider or publishing a service offering. You can find instructions for these basic operations in the Cloud Service Management Console online help.

Steps to configure resource pool

1. Download the content from HPLN : https://hpln.hpe.com/contentoffering/resource-provider-selection3
2. Import the content using Capsule Installer into CSA instance
3. Open the Resources tab and create a vCenter provider.

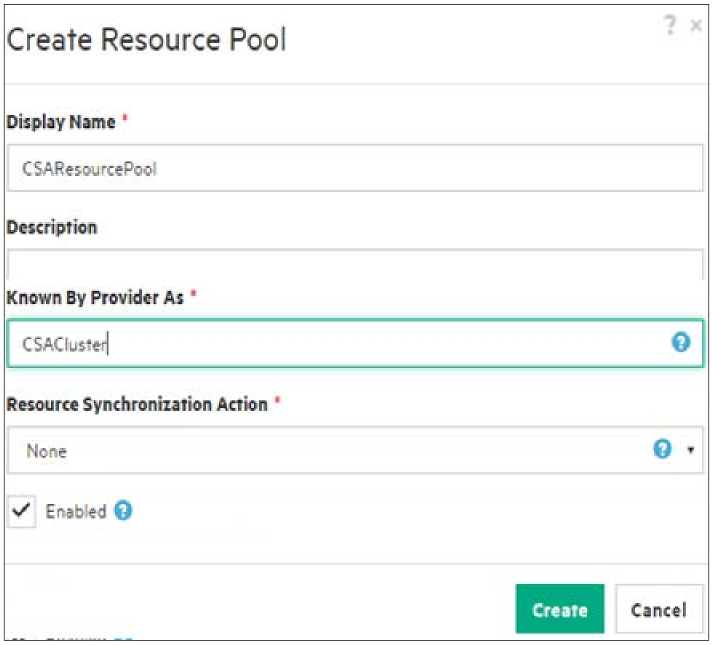
Figure 30 VCenter Resource Provider



4. Add a resource pool.

Click on the resource provider created to open the provider screen. Create a resource pool from Resource Pools tab. While creating the provider in the Known By Provider As field, add either the data center name or cluster name of the provider.

Figure 31. VCenter Resource Pool



Create Resource Pool

Display Name *
CSAResourcePool

Description

Known By Provider As *
CSACluster

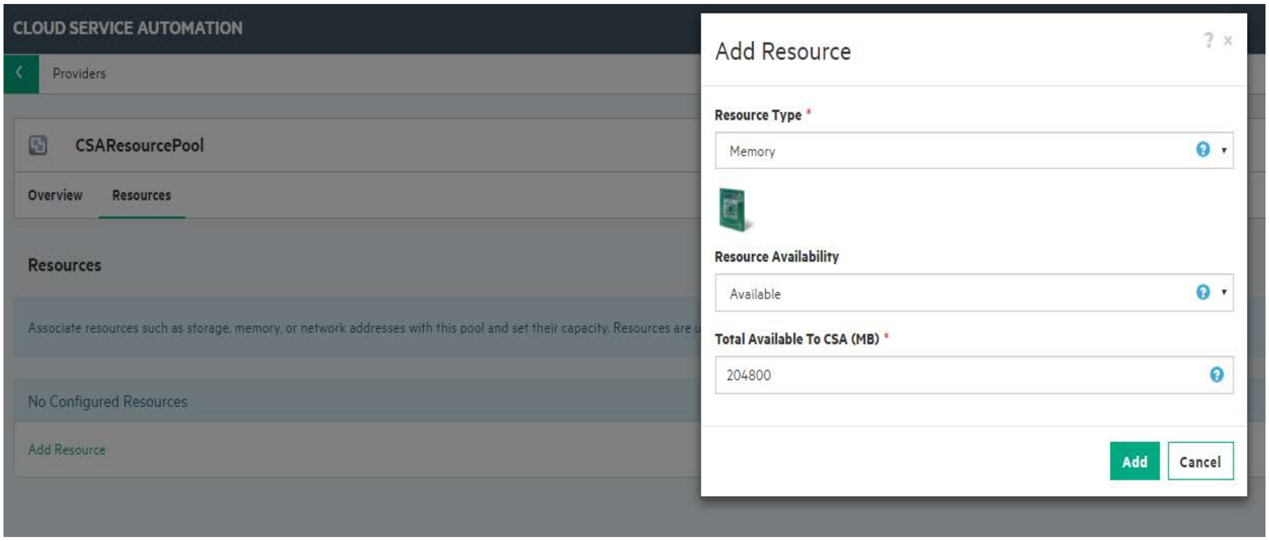
Resource Synchronization Action *
None

☑ Enabled

5. Add Resource Type to the Pool.

   Let us consume CPU and memory resources from the pool. Let us add these two resource types from the Resources tab on the pool. Add the Total Available to CSA values for the respective resource types. The Total Available to CSA value is the total capacity allocated for the resource that will be utilized.

   Figure 32. Add Resource types



Associate resources such as storage, memory, or network addresses with this pool and set their capacity. Resources are u

Add Resource

Resource Type *: Memory

Resource Availability: Available

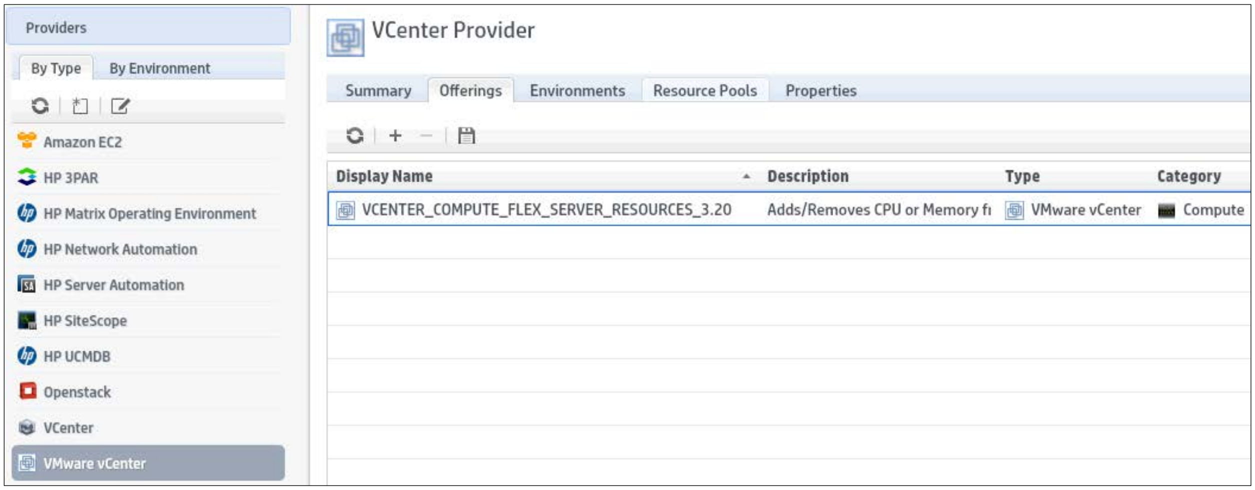
Total Available To CSA (MB) *: 204800

   We have two resource types CPU and memory added to the pool.

6. Associate resource offering to the provider.

   Add VCENTER_MODIFY_SERVER_RESOURCES_3.20 offering to vCenter provider

   Figure 33. Associate resource offering to provider



Display Name	Description	Type	Category
VCENTER_COMPUTE_FLEX_SERVER_RESOURCES_3.20	Adds/Removes CPU or Memory fi	VMware vCenter	Compute

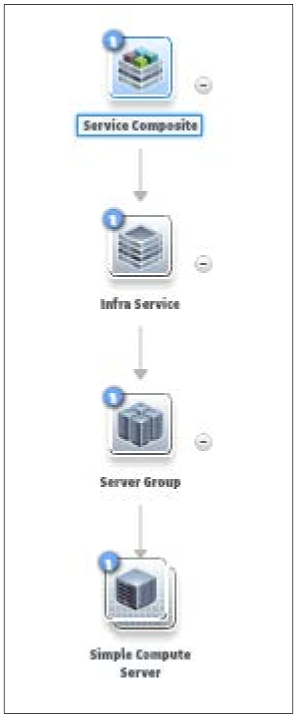
7. Create a service design.

   Let us create a service design with the following hierarchy of service components:

   Service Composite > Infra Service > Server Group > Server(Pattern).

   The server component is marked as pattern component.

Figure 34. Sample service design

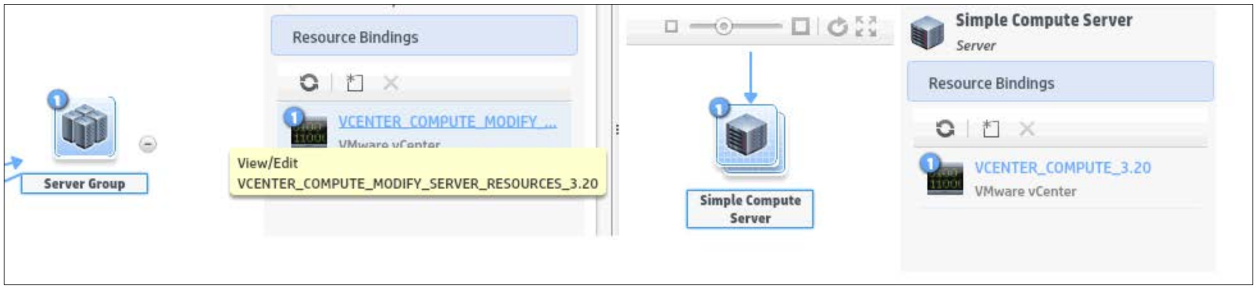


8. Configure resource offerings on components.

The following table lists down the resource offerings added on the components.

Component	Resource offering
Server Group	VCENTER_COMPUTE_MODIFY_SERVER_RESOURCES_3.20
SERVER	VCENTER_COMPUTE_3.20

Figure 35. Resource offering on component

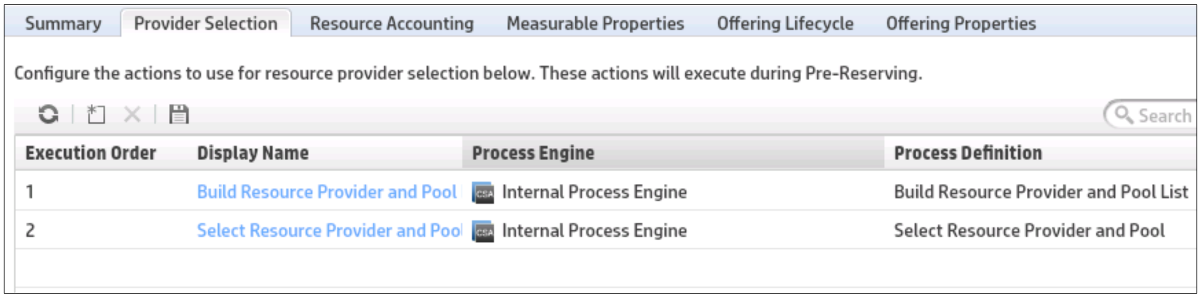


9. Configure provider and pool internal actions for the server group.

Add the following internal actions as shown in the table. These internal actions are added in the provider selection tab for the resource binding. Also add the provider property name on the internal actions as defined in the table below.

Component	Internal action	Provider property name
Server Group	Build Resource Provider and Pool List	
	Select Resource Provider and Pool	VCENTER_POOL
SERVER	Select Resource Provider and Pool from Parent	VCENTER_POOL

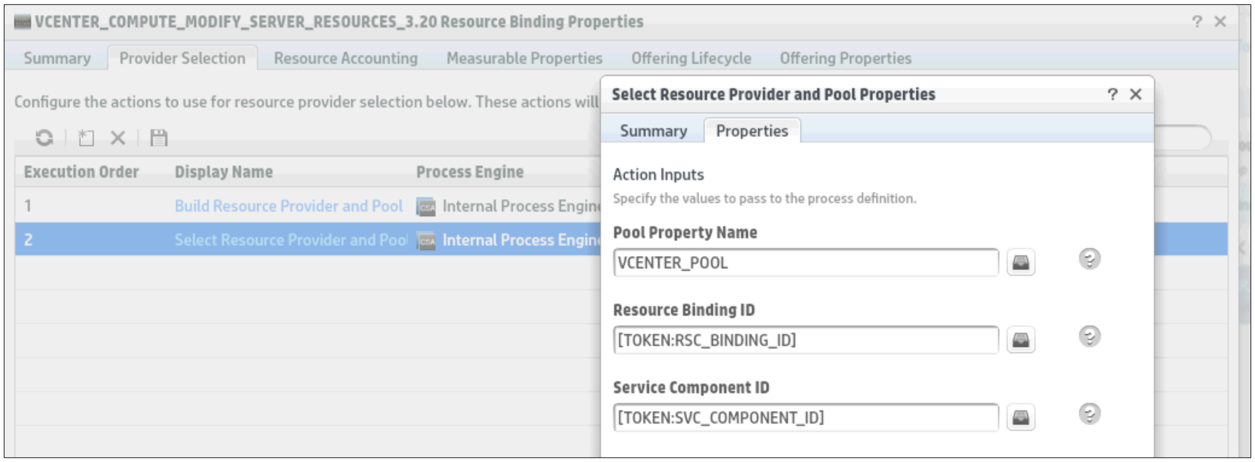
Figure 36.Internal actions on the server group resource binding (see step 7).



Configure the actions to use for resource provider selection below. These actions will execute during Pre-Reserving.

Execution Order	Display Name	Process Engine	Process Definition
1	Build Resource Provider and Pool	Internal Process Engine	Build Resource Provider and Pool List
2	Select Resource Provider and Pool	Internal Process Engine	Select Resource Provider and Pool

Figure 37. Setting provider properties internal actions on the server group resource binding



Configure the actions to use for resource provider selection below. These actions will

Execution Order	Display Name	Process Engine
1	Build Resource Provider and Pool	Internal Process Engine
2	Select Resource Provider and Pool	Internal Process Engine

Action Inputs

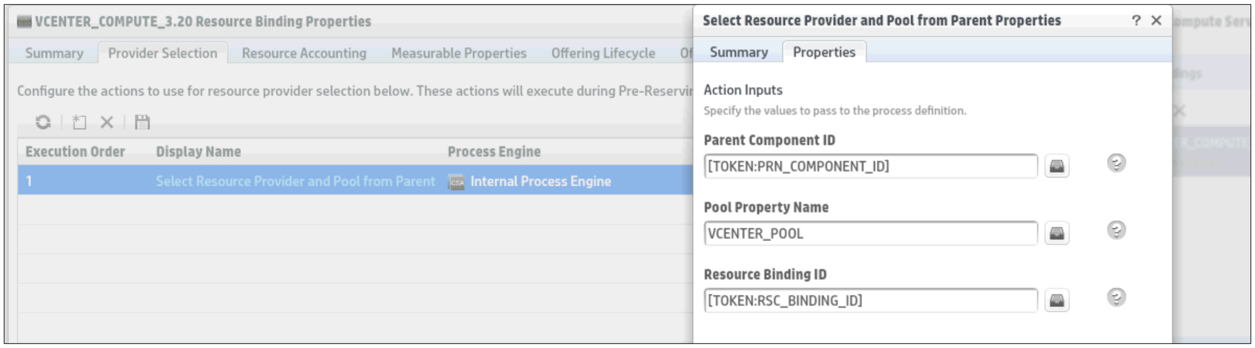
Specify the values to pass to the process definition.

Pool Property Name: VCENTER_POOL

Resource Binding ID: [TOKEN:RSC_BINDING_ID]

Service Component ID: [TOKEN:SVC_COMPONENT_ID]

Figure 38. Provider Property name on the Select Resource Provider and Pool from Parent internal action


VCENTER_COMPUTE_3.20 Resource Binding Properties
Summary
Provider Selection
Resource Accounting
Measurable Properties
Offering Lifecycle
Select Resource Provider and Pool from Parent Properties
Summary
Properties

Configure the actions to use for resource provider selection below. These actions will execute during Pre-Reservir

Execution Order	Display Name	Process Engine
1	Select Resource Provider and Pool from Parent	Internal Process Engine

Action Inputs

Specify the values to pass to the process definition.

Parent Component ID: [TOKEN:PRN_COMPONENT_ID]

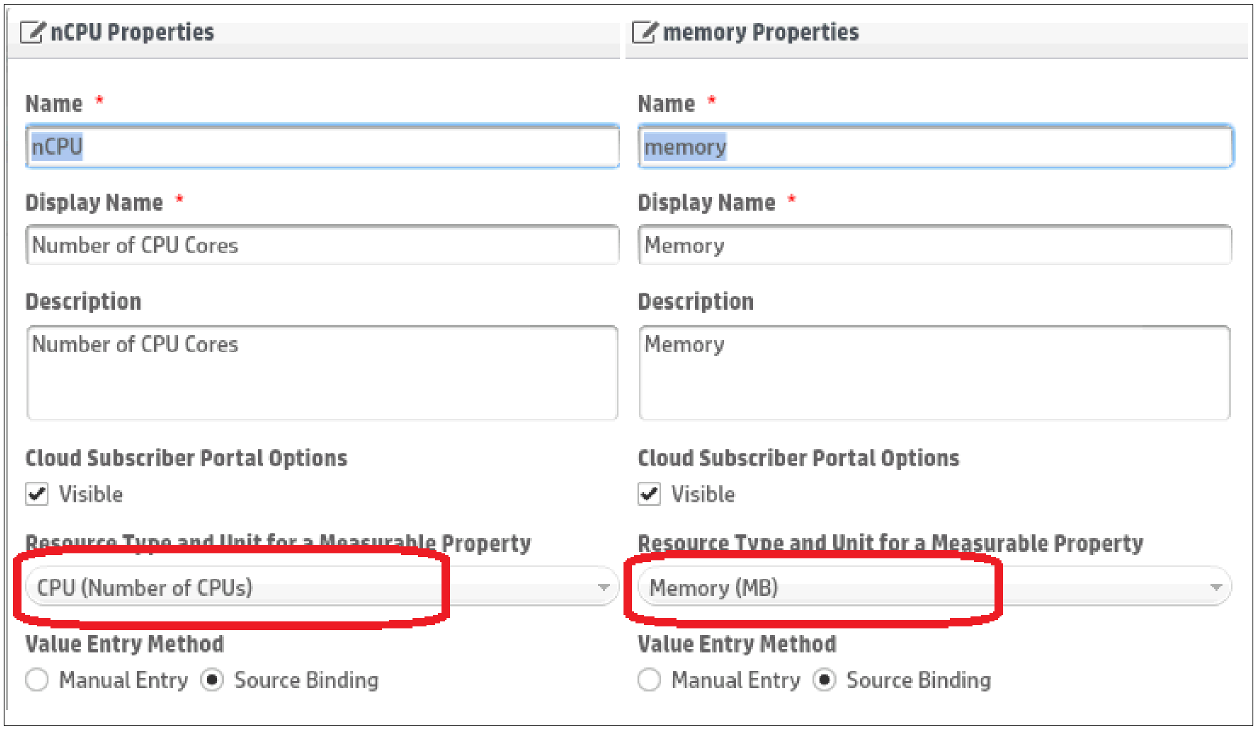
Pool Property Name: VCENTER_POOL

Resource Binding ID: [TOKEN:RSC_BINDING_ID]

10. Configure resource types on CPU and Memory properties.

    Open the Properties tab, where nCPU and memory are properties defined for the server component. These properties should be marked with the appropriate resource type as shown in Figure 39. The resource type on the properties qualifies them to be measurable.

Figure 39. Measurable properties defined on the component



nCPU Properties

Name *: nCPU

Display Name *: Number of CPU Cores

Description: Number of CPU Cores

Cloud Subscriber Portal Options: ☑ Visible

Resource Type and Unit for a Measurable Property: CPU (Number of CPUs)

Value Entry Method: ○ Manual Entry ● Source Binding

memory Properties

Name *: memory

Display Name *: Memory

Description: Memory

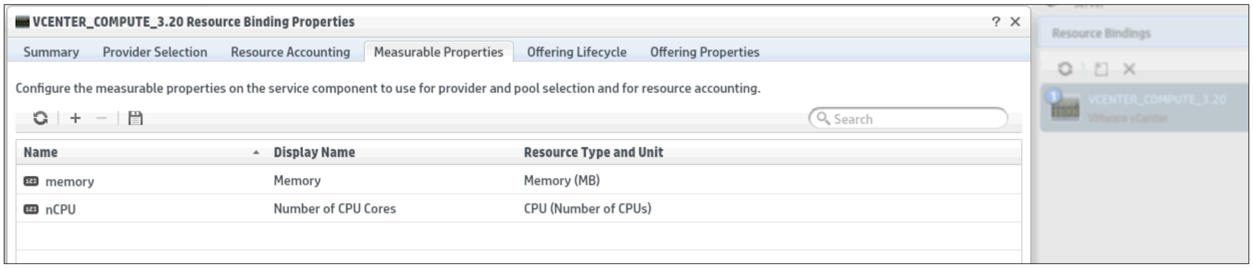
Cloud Subscriber Portal Options: ☑ Visible

Resource Type and Unit for a Measurable Property: Memory (MB)

Value Entry Method: ○ Manual Entry ● Source Binding

11. Configure nCPU and memory properties onto resource binding. Click Resource Binding VCENTER_COMPUTE_3.20 on the Server Component. Add the properties nCPU and memory onto Measurable Properties tab. The property values will be accounted for during provider and pool selection.

Figure 40. Add measurable properties



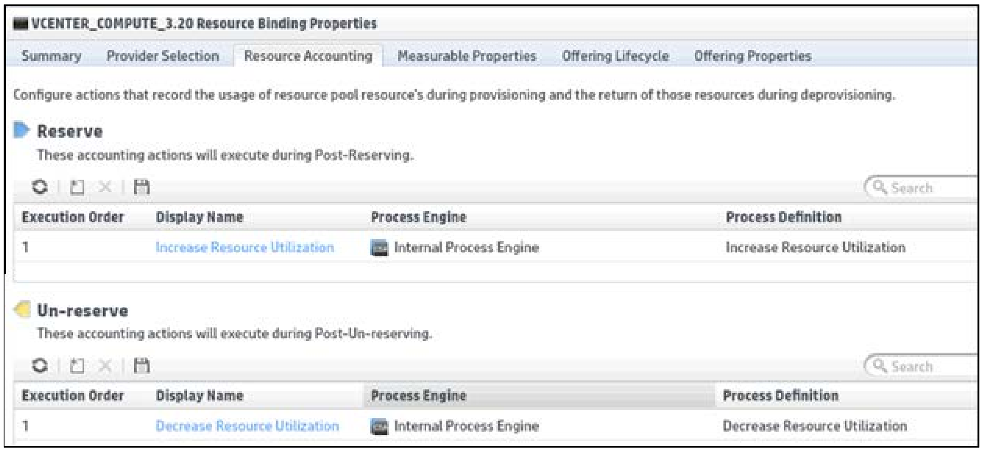
VCENTER_COMPUTE_3.20 Resource Binding Properties

Configure the measurable properties on the service component to use for provider and pool selection and for resource accounting.

Name	Display Name	Resource Type and Unit
memory	Memory	Memory (MB)
nCPU	Number of CPU Cores	CPU (Number of CPUs)

12. Configure resource accounting internal actions into resource binding. Add the internal action Increase Resource Utilization to the Reserve state and Decrease Resource Utilization to the Un-reserve state. These internal actions will be executed during the server component's Reserving/Un-reserving lifecycle execution phase.

Figure 41. Accounting actions on provider property name on Select Resource Provider and Pool from Parent internal action



VCENTER_COMPUTE_3.20 Resource Binding Properties

Configure actions that record the usage of resource pool resource's during provisioning and the return of those resources during deprovisioning.

Reserve

These accounting actions will execute during Post-Reserving.

Execution Order	Display Name	Process Engine	Process Definition
1	Increase Resource Utilization	Internal Process Engine	Increase Resource Utilization

Un-reserve

These accounting actions will execute during Post-Un-reserving.

Execution Order	Display Name	Process Engine	Process Definition
1	Decrease Resource Utilization	Internal Process Engine	Decrease Resource Utilization

13. Define property values on the components.

    Mandatory property values have to be defined for component properties for both Server group and Server components.

After these steps are executed, the service is ready to be associated with a service offering, and published to a catalog. After a user subscribes to this service, CSA creates a service instance which moves through the lifecycle phases.

Topology design use case

Use case: Resource pool

Administrators can configure resource quotas in CSA. The resource pool for a provider represents the resource capacity for each resource and configures them in the pool.

If an administrator wants to impose quotas on CPU and memory resources for cloud consumption, these resources must be consumed during provisioning of a cloud service and released during de- provisioning. Let's use topology designs to demonstrate the supply and demand of these resources.

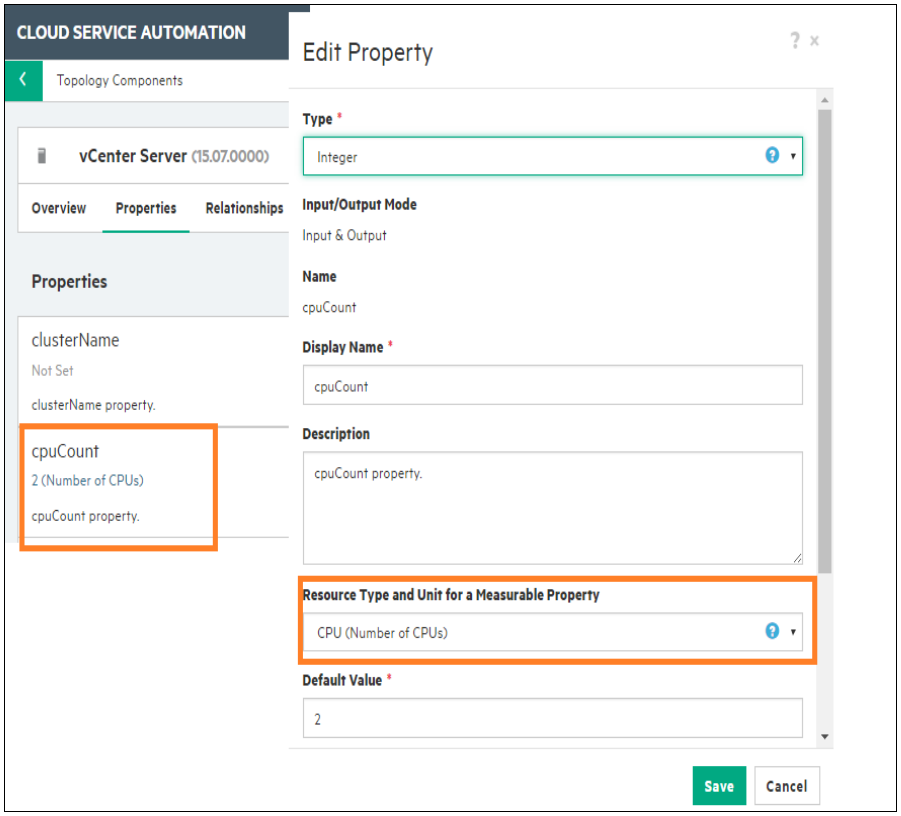
You can create topology designs for orderable services. Topology components belong to a provider type. Properties of a topology component can be marked as measurable to consume a resource capacity like CPU or memory. Using such components in a topology design will allow it to consume and release resources from the pool for cloud subscriptions.

Let us use a vCenter component in a topology design to demonstrate resource capacity and utilization management.

To configure resource capacity and utilization management for a provider, complete the following steps:

1. Create a resource provider for vCenter.
2. Define resource pool and resource capacities for the provider, such as CPU and memory.
3. Open the topology component and set the corresponding properties as measurable, such as cpuCount and memorySize, as shown in Figure 43.

   Figure 42. Setting a topology component property as measurable



Properties

clusterName
Not Set
clusterName property.

cpuCount
2 (Number of CPUs)
cpuCount property.

Edit Property

Type *: Integer

Input/Output Mode: Input & Output

Name: cpuCount

Display Name *: cpuCount

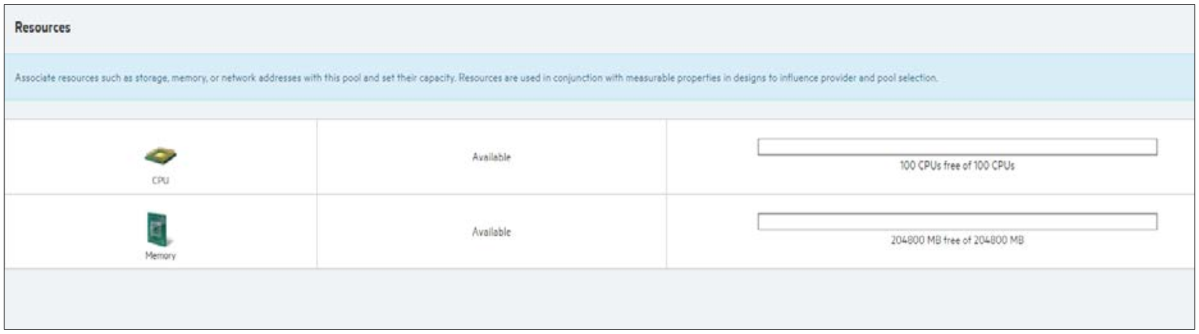
Description: cpuCount property.

Resource Type and Unit for a Measurable Property: CPU (Number of CPUs)

Default Value *: 2

4. Use this component in a service design and enter values for mandatory fields, such as:
   - cpuCount = 2
   - memorySize=2048
   - customizationSpec = <VCenter customization specification file>
   - vmNamePrefix=<Name of the VM>
   - vmTemplateReference=<Clone Template name in VCenter>
5. Save and publish the service design to create a service offering. Issue a subscription for this offering.
6. Verify that resources have been consumed from the vCenter resource pool, as shown in Figure 44.

Figure 43. Resources consumed from a resource pool



Resources

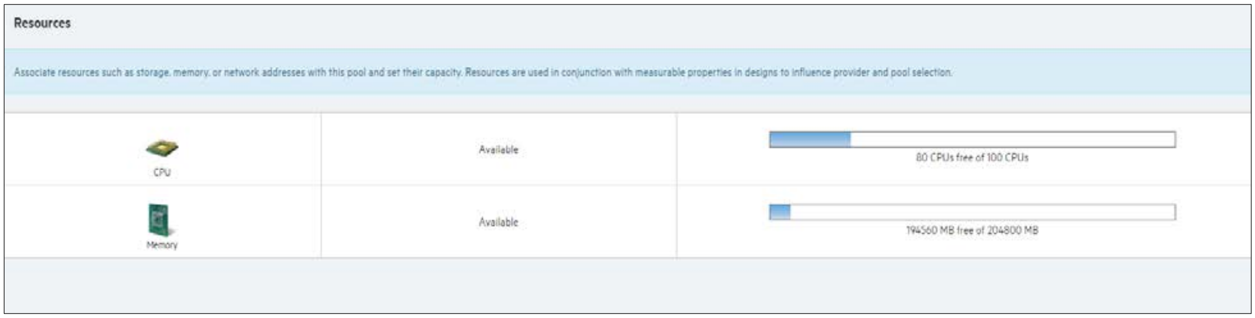
Associate resources such as storage, memory, or network addresses with this pool and set their capacity. Resources are used in conjunction with measurable properties in designs to influence provider and pool selection.

CPU	Available	100 CPUs free of 100 CPUs
Memory	Available	204800 MB free of 204800 MB

Resource utilization report

Administrators can view resource capacity allocated and resources utilized for each resource pool from the Cloud Service Management Console. You can increase the resource capacity allocated manually. Automatic resource capacity allocation can be set using the Resource Sync option for the resource pool.

Figure 44. Resource utilization report in the Cloud Service Management Console



Resources

Associate resources such as storage, memory, or network addresses with this pool and set their capacity. Resources are used in conjunction with measurable properties in designs to influence provider and pool selection.

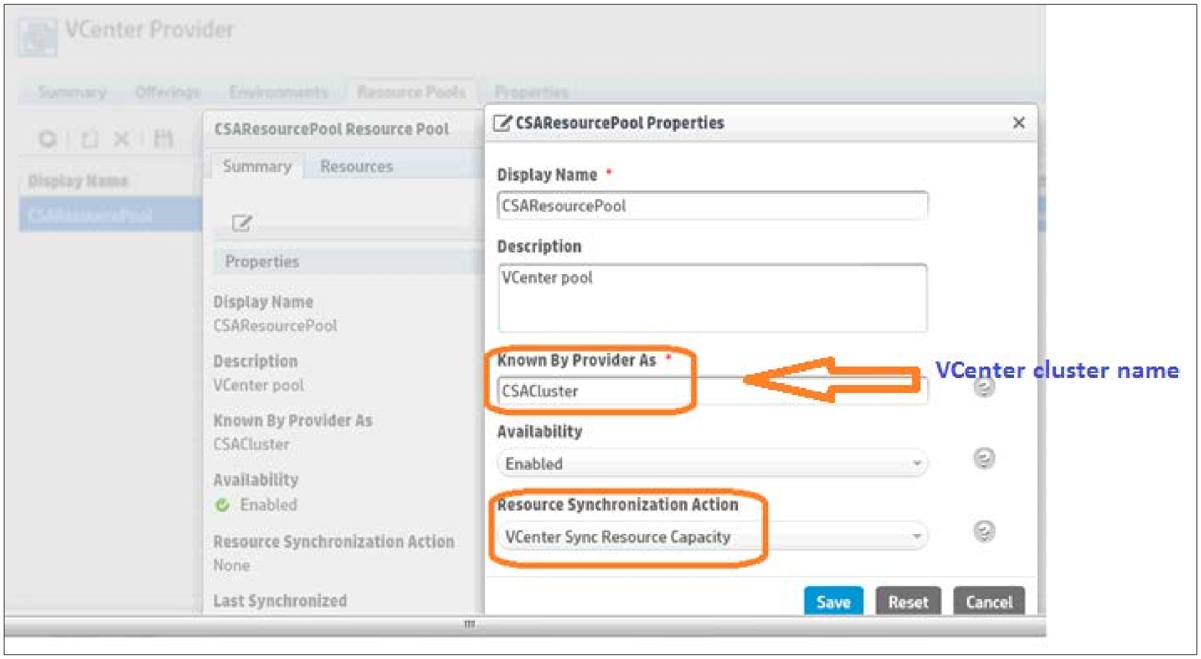
CPU	Available	80 CPUs free of 100 CPUs
Memory	Available	194560 MB free of 204800 MB

Resource synchronization

Resource capacity for the resource pool is used during provider and pool selection. Administrators can set the capacity manually or automatically. The Utilization REST API can be used to retrieve available resource capacity units from a provider's single hypervisor or cluster or a datacenter. The Utilization REST API can also be used to update the Total Available to CSA value. CSA can use OO flows to retrieve allocated units and update resource capacity on the pool. CSA provides the VCenter Sync Resource Capacity flow, which will update the resource capacity retrieved from a vCenter provider.

You can set the OO action flow using the Resource Synchronization Action option on the resource pool. While configuring the sync action, Known By Provider As should be a provider cluster name or a data center name.

Figure 45. vCenter Resource Synchronization action



Display Name
CSAResourcePool

Description
VCenter pool

Known By Provider As
CSACluster

Availability
Enabled

Resource Synchronization Action
None

Last Synchronized

CSAResourcePool Properties

Display Name *
CSAResourcePool

Description
VCenter pool

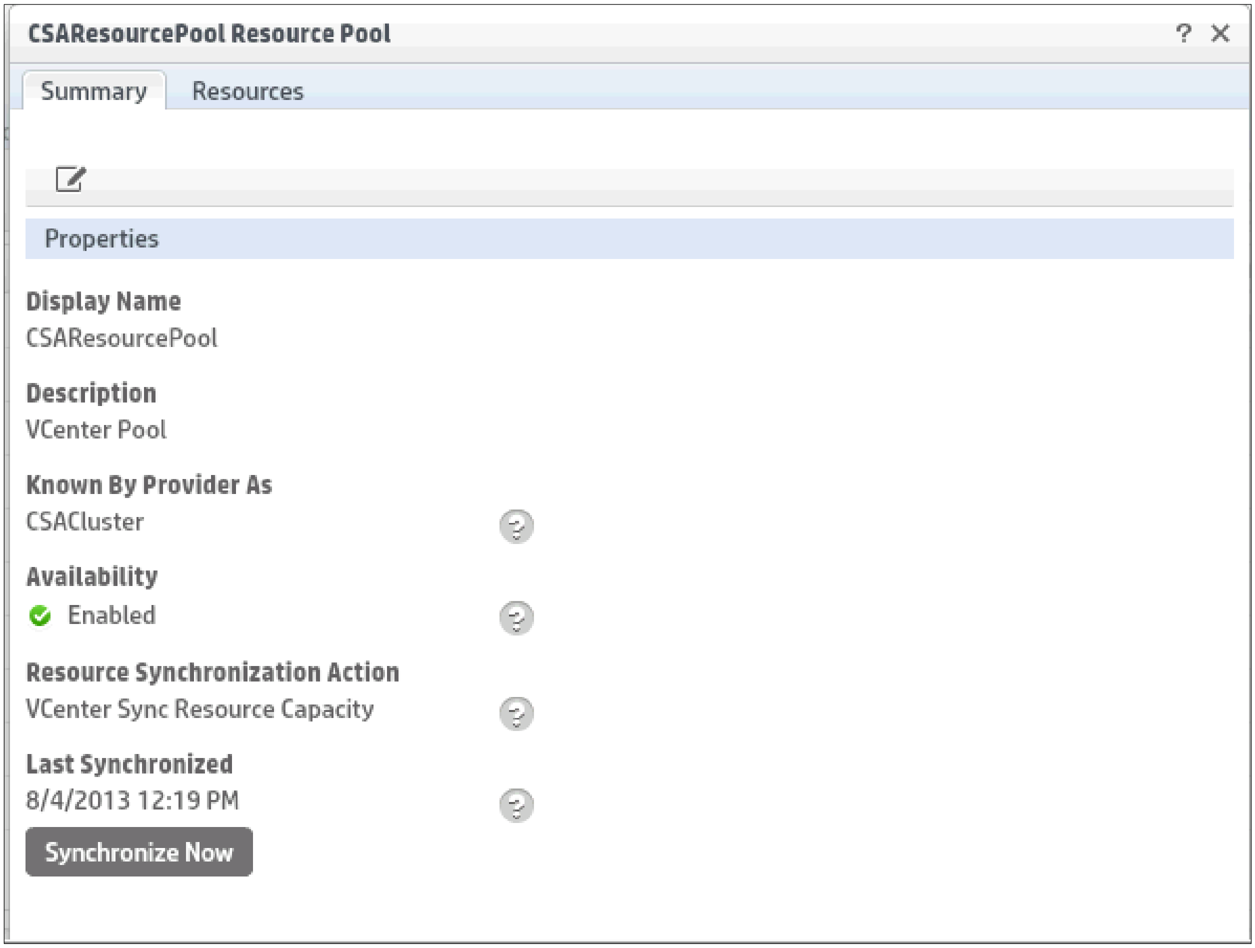
Known By Provider As *
CSACluster

Availability
Enabled

Resource Synchronization Action
VCenter Sync Resource Capacity

Resource synchronization can be scheduled by triggering the Synchronize Now button for the resource pool. On success, the Last Synchronized field will display the latest timestamp when the resource pool was updated.

Figure 46. VCenter Resource Synchronization action



CSAResourcePool Resource Pool

Properties

Display Name
CSAResourcePool

Description
VCenter Pool

Known By Provider As
CSACluster

Availability
Enabled

Resource Synchronization Action
VCenter Sync Resource Capacity

Last Synchronized
8/4/2013 12:19 PM

Appendix A: REST APIs for custom provider selection

Get the list of candidate providers

To get the list of candidate providers, use the GET artifact API.

URL	`https://localhost:8444/csa/rest/artifact/`**`<Resource_Binding_ID>`**`?userIdentifier=<UserID>&scope=view&view=candidateproviders`
Method	**GET**

Response	
	```
	<ResourceBinding>
	     <id>90e72e4f3b622418013b6d1cde3e0162</id>
	     <objectId>90e72e4f3b622418013b6d1cde3e0162</objectId>
	     <isCriticalSystemObject>false</isCriticalSystemObject>
	     <description>binding_December 5, 2012 7:22:43 PM UTC</description>
	     <name>binding_December 5, 2012 7:22:43 PM UTC</name>
	     <displayName>binding_December 5, 2012 7:22:43 PM UTC</displayName>
	     ......
	     <candidateProvider>
	          <id>90e72e4f3b622418013b686efbd200c7</id>
	          <objectId>90e72e4f3b622418013b686efbd200c7</objectId>
	          <isCriticalSystemObject>false</isCriticalSystemObject>
	```

```
          <name>Provider1_December 5, 2012 12:20:01 AM UTC</name>
          <displayName>Provider1</displayName>
          <property>
               <id>90e72e4f3b622418013b6df6fc440210</id>
               <objectId>90e72e4f3b622418013b6df6fc440210</objectId>
               <isCriticalSystemObject>false</isCriticalSystemObject>
               <name>ORGANIZATIONS</name>
               <displayName>Organizations</displayName>
               <values>
                    <value>A</value>
               </values>
               <maxOccurs>1</maxOccurs>
               <minOccurs>1</minOccurs>
               <orderIndex>0</orderIndex>
               <confidential>false</confidential>
               <encrypted>false</encrypted>
               <consumerReadOnly>true</consumerReadOnly>
              <consumerVisible>false</consumerVisible>
          </property>
          <disabled>false</disabled>
          <numberOfResourceOffering>0</numberOfResourceOffering>
          <numberOfEnvironment>0</numberOfEnvironment>
          <numberOfPools>0</numberOfPools>
     </candidateProvider>
     <candidateProvider> ... </candidateProvider>
     <candidateProvider> ... </candidateProvider>
</ResourceBinding>
```

Update the list of valid providers

To update the list of valid providers, use the PUT artifact API to include the refined list of valid providers. If the GET call provides three candidate providers, the PUT call can update the list to two valid providers. The lifecycle engine will then choose one provider from these two valid providers.

URL	https://localhost:8444/csa/rest/artifact/**<RESOURCE_BINDING_ID>**?userIdentifier=<User_ID>&scope=view&view=validproviders
Method	**PUT**

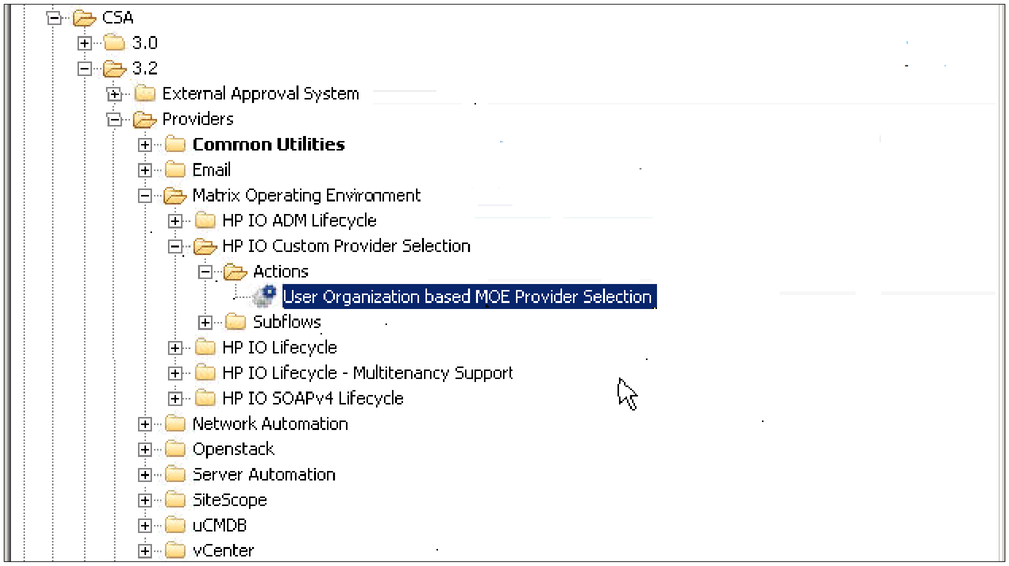
Payload	<ResourceBinding>
	     <id>90e72e243992d89e013992fc10370056</id>
	     <validProvider>
	          <id>90e72e4f3b622418013b686efbd200c7</id>
	     validProvider>
	     <validProvider>
	          <id>90e72e4f3b622418013b686f5b5e00ca</id>
	     </validProvider>
	</ResourceBinding>

Response	<ResourceBinding>
	     <id>90e72e4f3b622418013b6d1cde3e0162</id>
	     <objectId>90e72e4f3b622418013b6d1cde3e0162</objectId>
	     <isCriticalSystemObject>false</isCriticalSystemObject>
	     <name>binding_December 5, 2012 7:22:43 PM UTC</name>
	     <displayName>binding_December 5, 2012 7:22:43 PM UTC</displayName>
	     ......
	     <validProvider>
	          <id>90e72e4f3b622418013b686efbd200c7</id>
	          <objectId>90e72e4f3b622418013b686efbd200c7</objectId>
	          <isCriticalSystemObject>false</isCriticalSystemObject>
	          <name>Provider1_December 5, 2012 12:20:01 AM UTC</name>
	          <displayName>Provider1</displayName>
	          <disabled>false</disabled>
	          <numberOfResourceOffering>0</numberOfResourceOffering>
	          <numberOfEnvironment>0</numberOfEnvironment>
	          <numberOfPools>0</numberOfPools>
	     </validProvider>
	     <validProvider>
	          <id>90e72e4f3b622418013b686f5b5e00ca</id>
	          <objectId>90e72e4f3b622418013b686f5b5e00ca</objectId>
	          <isCriticalSystemObject>false</isCriticalSystemObject>
	          <name>Provider2_December 5, 2012 12:20:26 AM UTC</name>
	          <displayName>Provider2</displayName>
	          <disabled>false</disabled>
	          <numberOfResourceOffering>0</numberOfResourceOffering>
	          <numberOfEnvironment>0</numberOfEnvironment>
	          <numberOfPools>0</numberOfPools>
	     </validProvider>
	</ResourceBinding>

Appendix B: HPE Operations Orchestration flow for custom provider selection

This section provides a brief overview of how custom provider selection is implemented in the User Organization based MOE Provider Selection flow. This flow is provided as sample content with CSA. Refer to the *CSA Configuration Guide* for information on how to configure OO to obtain the flows.

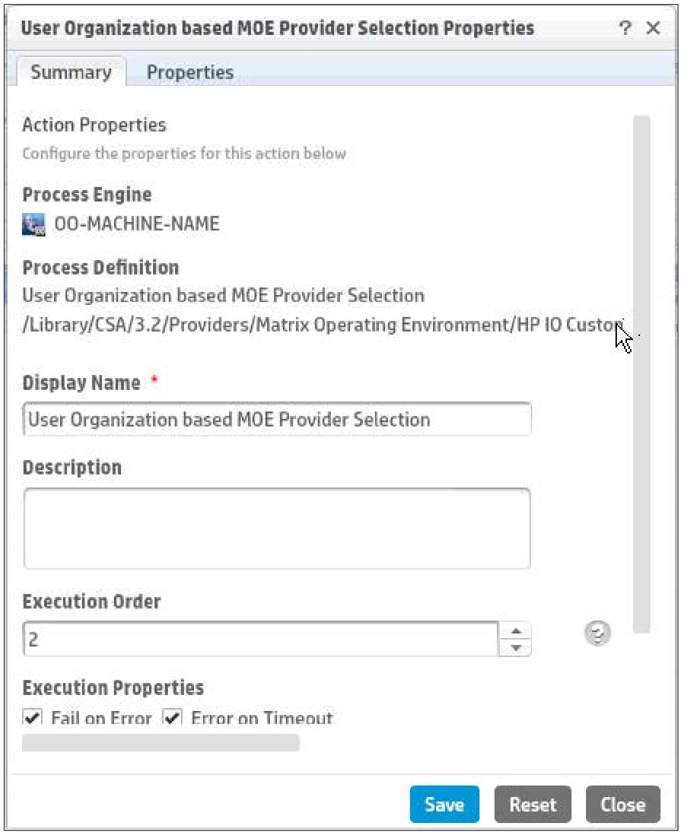
Figure 47. Custom provider selection flow for Matrix OE



- CSA
  - 3.0
  - 3.2
    - External Approval System
    - Providers
      - **Common Utilities**
      - Email
      - Matrix Operating Environment
        - HP IO ADM Lifecycle
        - HP IO Custom Provider Selection
          - Actions
            - User Organization based MOE Provider Selection
          - Subflows
        - HP IO Lifecycle
        - HP IO Lifecycle - Multitenancy Support
        - HP IO SOAPv4 Lifecycle
      - Network Automation
      - Openstack
      - Server Automation
      - SiteScope
      - uCMDB
      - vCenter

Figure 49 shows the CSA action used in the Matrix OE design for performing custom provider selection. The Process Definition field shows the details of the corresponding flow.

Figure 48. CSA action for custom provider selection



User Organization based MOE Provider Selection Properties

Action Properties

Configure the properties for this action below

Process Engine

OO-MACHINE-NAME

Process Definition

User Organization based MOE Provider Selection

/Library/CSA/3.2/Providers/Matrix Operating Environment/HP IO Custon

Display Name *

User Organization based MOE Provider Selection

Description

Execution Order

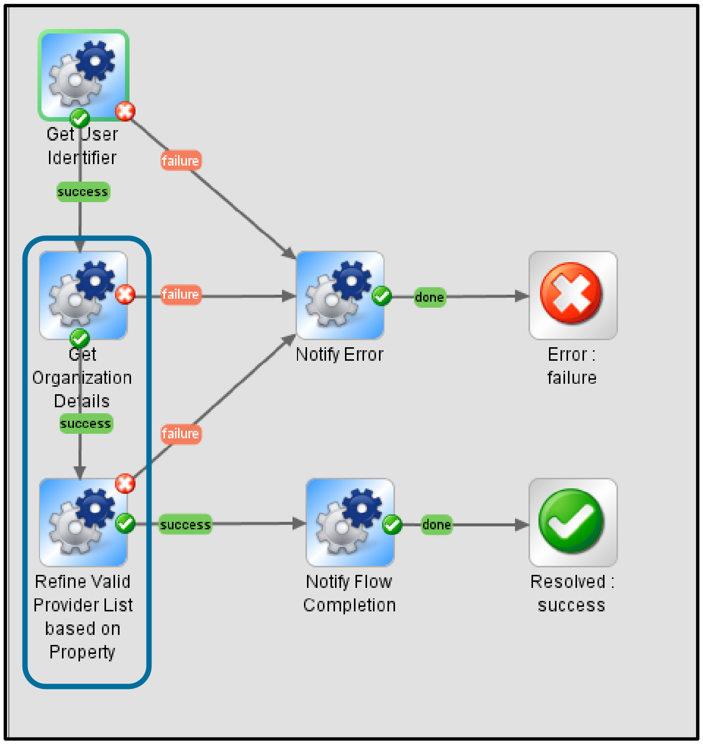
2

Execution Properties

☑ Fail on Error ☑ Error on Timeout

Figure 50 shows the flow that is invoked from CSA corresponding to the action shown in Figure 49. The steps which implement the refinement logic are shown in the blue box in Figure 50. For this flow, the refinement condition is based on the requesting user's organization. Each resource provider has a CSA property called ORGANIZATIONS which specifies a list of organizations. A provider will be included in the list of valid providers only if its ORGANIZATIONS property includes the requesting user's organization.

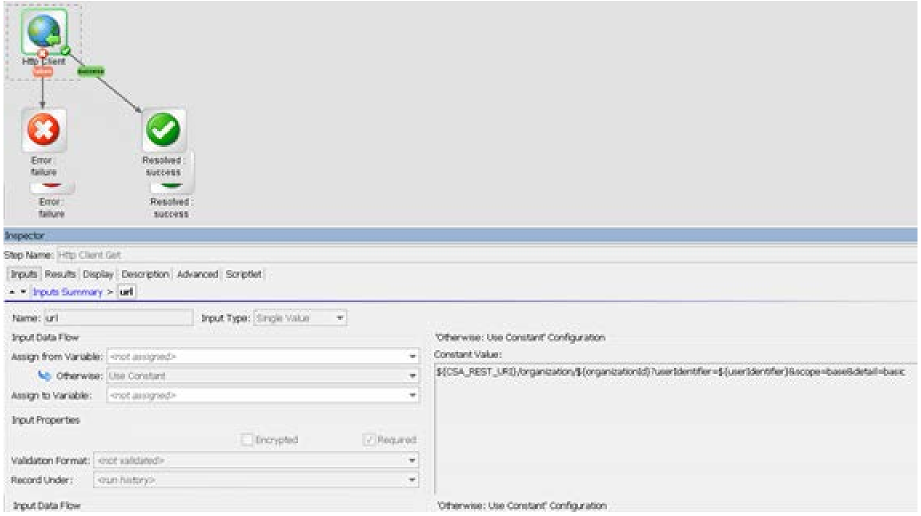
Figure 49. User organization-based Matrix OE provider selection flow



Get organization details

This step gets the requesting user's organization (organization name). This is implemented by using the Organization REST API from CSA. Figure 51 shows the URI used by the subflow that corresponds to this step.

Figure 50. Get Organization Details subflow



Step Name: Http Client Get

Name: url Input Type: Single Value

Input Data Flow

Assign from Variable: <not assigned>

Otherwise: Use Constant

Assign to Variable: <not assigned>

Input Properties

□ Encrypted ☑ Required

Validation Format: <not validated>

Record Under: <run history>

'Otherwise: Use Constant' Configuration

Constant Value:

${CSA_REST_URI}/organization/${organizationId}?userIdentifier=${userIdentifier}&scope=base&detail=basic

Input Data Flow

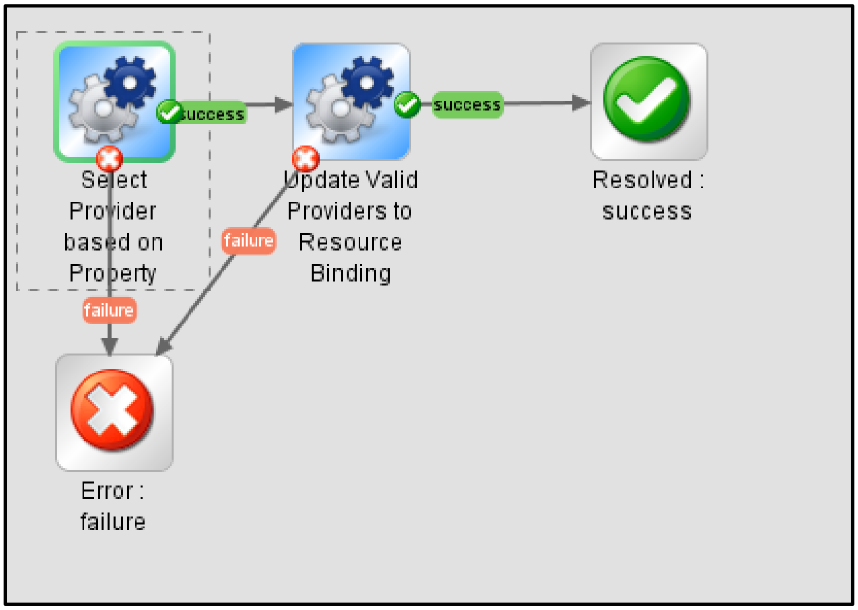
'Otherwise: Use Constant' Configuration

Refine valid provider list based on property

The requesting user’s organization value is used by this step to filter only those providers that can support the user’s organization. Figure 52 shows the subflow corresponding to this step:

- *Select Provider based on property:* This subflow step gets the list of candidate providers and filters the providers based on the value of their ORGANIZATIONS property.
- *Update Valid providers to Resource binding:* This subflow step updates the list of valid providers in CSA with the list built from the previous step.

Figure 51. Refine Valid Provider List based on Property subflow

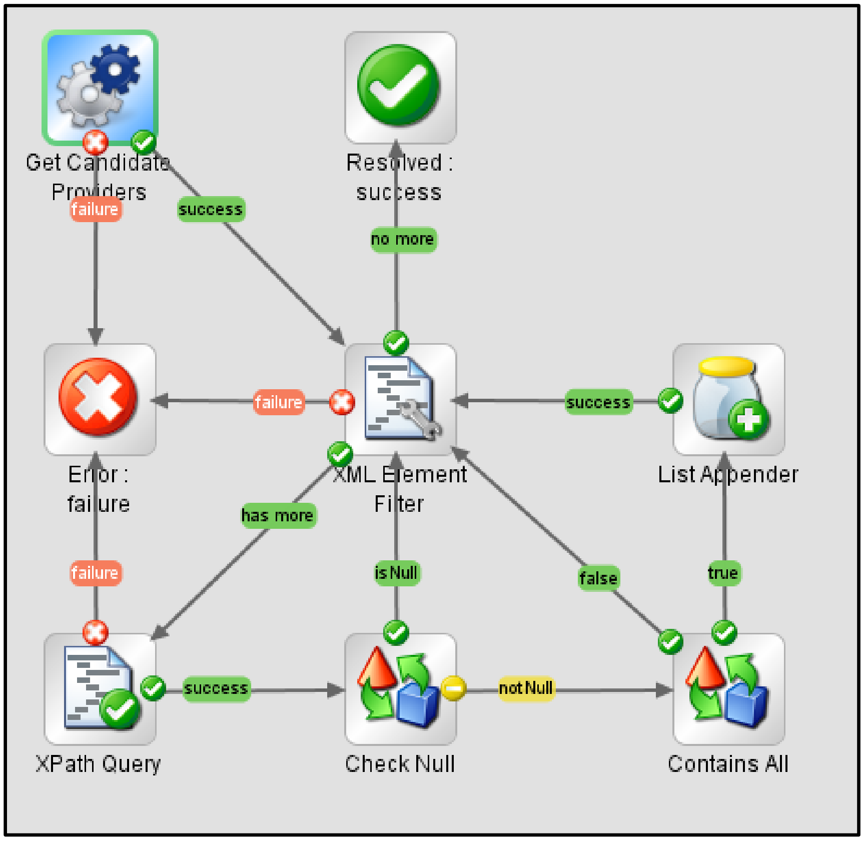

Select Provider based on Property
success
failure
Update Valid Providers to Resource Binding
success
failure
Resolved : success
Error : failure

Select provider based on property

The subflow corresponding to this step is shown in Figure 53 and has the following major steps:

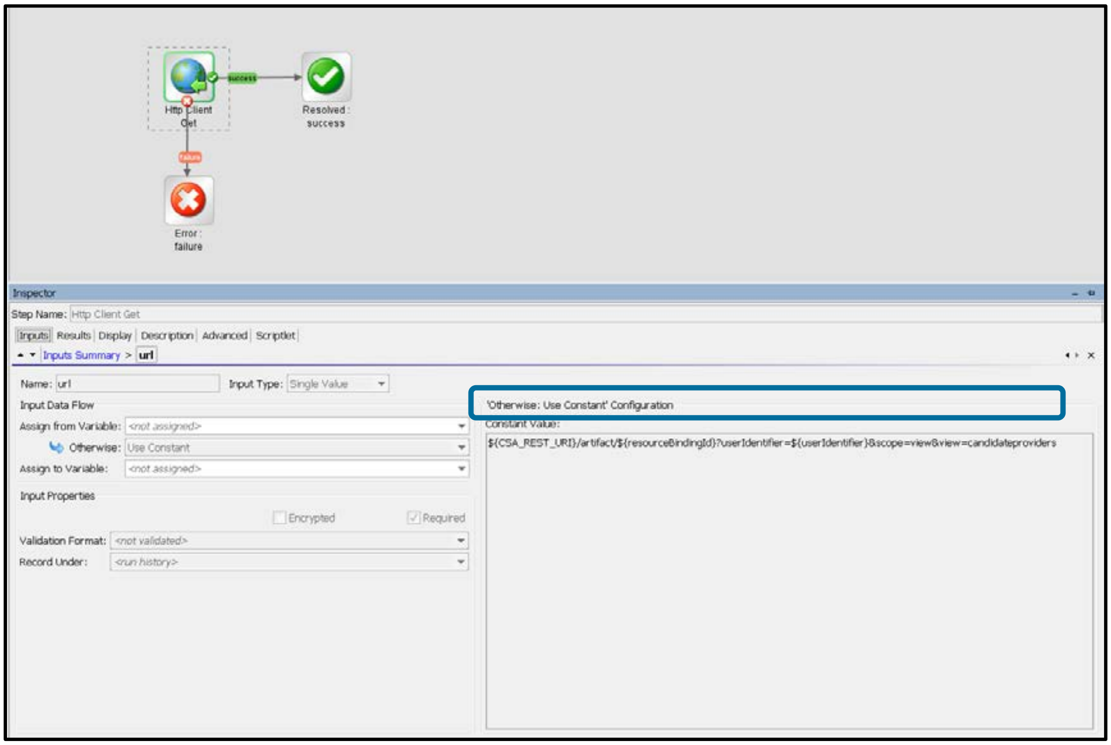
- *Get Candidate Providers:* This step uses the CSA Artifact REST API to get the list of candidate providers for the resource binding.
- *XML Element Filter:* This step looks for the ResourceBinding/candidateProvider element of the result from the previous step.
- *XPath Query:* For each candidate provider, this step queries the ORGANIZATIONS property value.
- *Contains All:* This step checks if the requesting user’s organization value is contained in the ORGANIZATIONS property of the resource provider.
- *List Appender:* This step appends the resource provider to the list of valid resource providers if the previous step returns true (when the resource provider’s ORGANIZATIONS property has the requesting user’s organization name).

Figure 52. Select Provider Based on Property subflow



The Get Candidate Providers subflow uses the HTTP Client GET operation. Figure 54 shows the CSA REST URI used with the candidateproviders view.

Figure 53. Get Candidate Providers subflow



Step Name: Http Client Get

Name: url Input Type: Single Value

Input Data Flow

Assign from Variable: <not assigned>

Otherwise: Use Constant

Assign to Variable: <not assigned>

Input Properties

□ Encrypted ☑ Required

Validation Format: <not validated>

Record Under: <run history>

'Otherwise: Use Constant' Configuration

Constant Value:

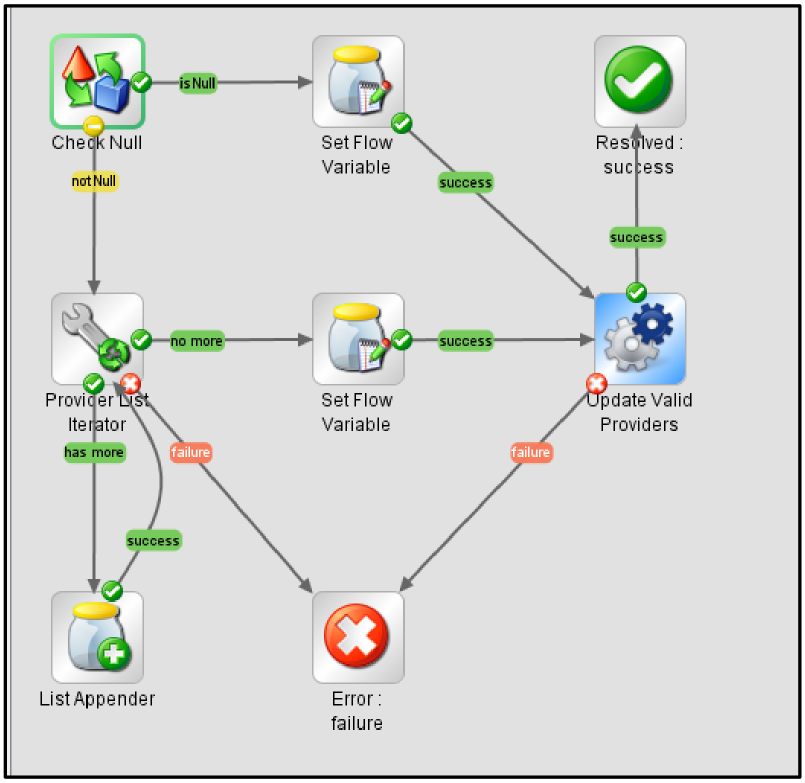
```
${CSA_REST_URI}/artifact/${resourceBindingId}?userIdentifier=${userIdentifier}&scope=view&view=candidateproviders
```

Update valid providers to resource binding

The subflow corresponding to this step is shown in Figure 55 and has the following major steps:

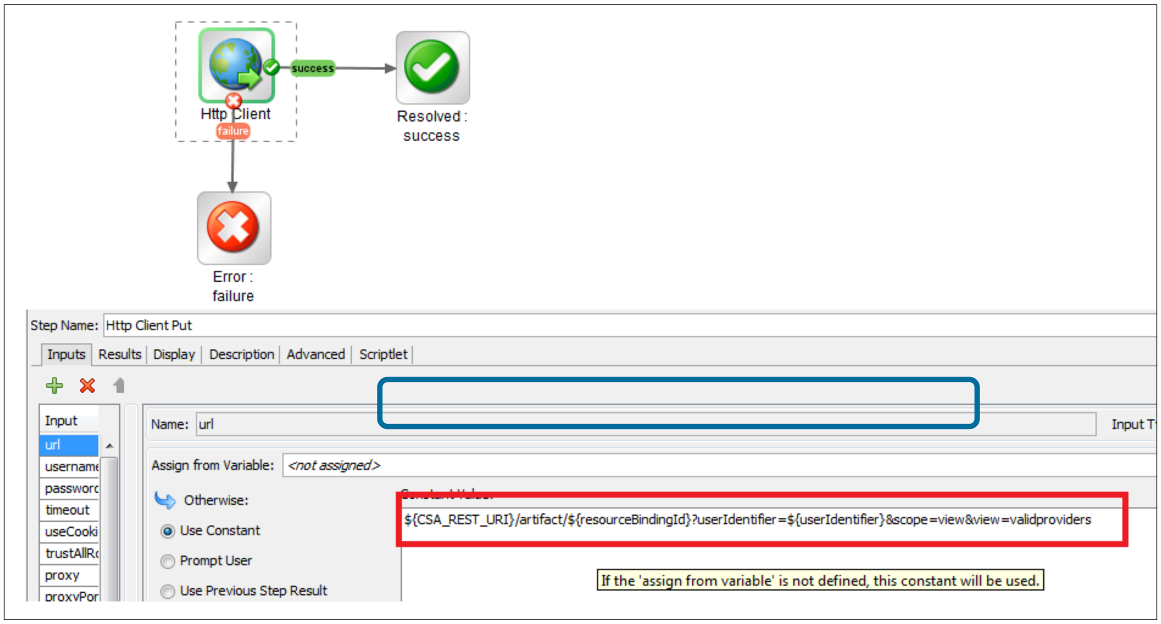
- *Provider List Iterator* and *List Appender:* Together these two steps iterate over each of the valid providers filtered by the Select Providers based on Property subflow and create the body of the PUT request to update the resource binding.
- *Update Valid Providers:* This step uses the CSA Artifact REST API to update the list of valid providers for the resource binding.

Figure 54. Update Provider List to Resource Binding subflow



The Update Valid Providers subflow uses the HTTP Client PUT operation. Figure 56 shows the CSA REST API with the validproviders view, used by this operation to update the list of valid providers.

Figure 55. Update Valid Providers subflow



Step Name: Http Client Put

Name: url

Assign from Variable: *<not assigned>*

● Use Constant

○ Prompt User

○ Use Previous Step Result

```
${CSA_REST_URI}/artifact/${resourceBindingId}?userIdentifier=${userIdentifier}&scope=view&view=validproviders
```

If the 'assign from variable' is not defined, this constant will be used.

Appendix C: Resource providers and pools

Get the list of candidate providers and pools

To get the list of candidate providers and pools, use the GET artifact API. The candidate pool XML snippet is bundled within the canididatepool XML fragment.

URI	`https://localhost:8444/csa/rest/artifact/`**`<RESOURCE_BINDING_ID>`**`?userIdentifier=<USER_ID>&scope=view&view=candidatepools`
Method	**GET**

Response	`<ResourceBinding>`
	`     <id>90e72d893cd0fb1d013cd108d496004f</id>`
	`     <objectId>90e72d893cd0fb1d013cd108d496004f</objectId>`
	`     <isCriticalSystemObject>false</isCriticalSystemObject>`
	`     <description>binding_February 12, 2013 6:14:59 PM UTC</description>`
	`<name>binding_February 12, 2013 6:14:59 PM UTC</name>`
	`     <displayName>binding_February 12, 2013 6:14:59 PM UTC</displayName>`
	`     <artifactType>`
	`     ......`
	`     </artifactType>`
	`     <disabled>false</disabled>`
	`     <candidateProvider>`
	`          <id>90e72d893ccf9a39013ccf9cf167001a</id>`
	`          <resourceProvider>`
	`               <id>90e72d893ccf9a39013ccf9cf167001a</id>`
	`               <objectId>90e72d893ccf9a39013ccf9cf167001a</objectId>`
	`               <isCriticalSystemObject>false</isCriticalSystemObject>`
	`               <name>Provider2_February 12, 2013 6:13:54 PM UTC</name>`
	`               <displayName>Provider2</displayName>`
	`               <disabled>false</disabled>`
	`          </resourceProvider>`
	`          <candidatePool>`
	`               <id>90e72d893ccf9a39013ccf9d5d630022</id>`
	`               <objectId>90e72d893ccf9a39013ccf9d5d630022</objectId>`
	`               <isCriticalSystemObject>false</isCriticalSystemObject>`
	`               <name>P2-Pool1_February 12, 2013 6:14:22 PM UTC</name>`
	`               <displayName>P2-Pool1</displayName>`
	`               <disabled>false</disabled>`
	`               <useProviderEnv>false</useProviderEnv>`
	`          </candidatePool>`
	`          <candidatePool>`
	`          ......`
	`          </candidatePool>`
	`     </candidateProvider>`
	`</ResourceBinding>`

Update the list of valid providers

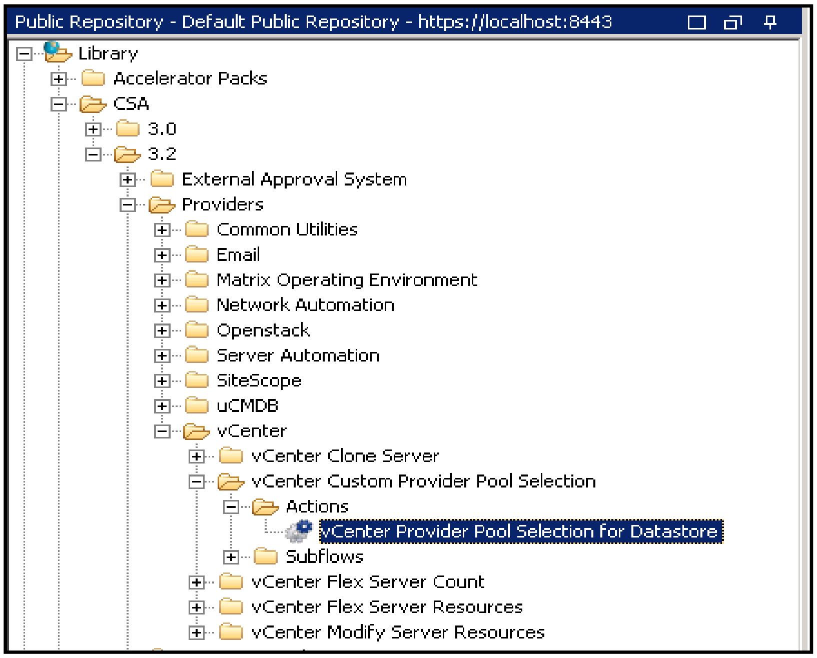
To update the list of valid providers, use the PUT artifact API.

<table>
<tr><td><b>URI</b></td><td>https://localhost:8444/csa/rest/artifact/**<RESOURCE_BINDING_ID>**?userIdentifier=<USER_ID>&scope=view&view=validproviderspools</td></tr>
<tr><td><b>Method</b></td><td><b>PUT</b></td></tr>
<tr><td><b>Payload</b></td><td>Include the refined list of valid providers and pool on the binding id.
<pre>
<ResourceBinding>
        <id>90e72d893ccf9a39013ccfa0076a0099</id>
        <validProvider>
                <resourceBinding>
                        <id>90e72d893ccf9a39013ccfa0076a0099</id>
                </resourceBinding>
                <resourceProvider>
                        <id>90e72d893ccf9a39013ccf9cf167001a</id>
                </resourceProvider>
                <validPool>
                        <id>90e72d893ccf9a39013ccf9c6be60014</id>
                </validPool>
                <validPool>
                        <id>90e72d893ccf9a39013ccf9c6be60016</id>
                </validPool>
        </validProvider>
        <validProvider>
        ......
        </validProvider>
</ResourceBinding>
</pre></td></tr>
</table>

Appendix D: HPE Operations Orchestration flows for custom providers and pools

This appendix provides a brief overview of how custom pool selection is implemented in the vCenter Provider Pool Selection for Datastore flow. This flow is provided with CSA. See *CSA Configuration Guide* for configuration steps for OO and how to obtain the flows from CSA.

Figure 56. Provider Pool Selection OO Flow for vCenter

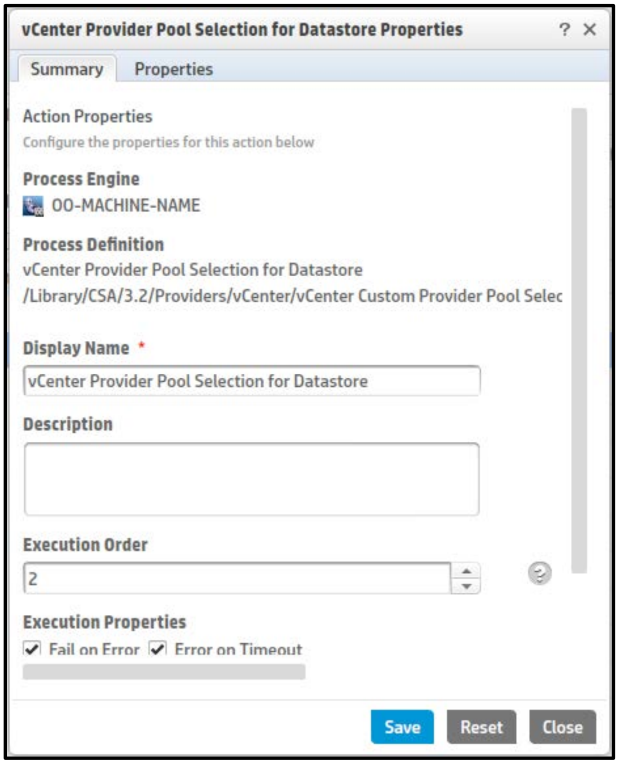


Public Repository - Default Public Repository - https://localhost:8443

- Library
  - Accelerator Packs
  - CSA
    - 3.0
    - 3.2
      - External Approval System
      - Providers
        - Common Utilities
        - Email
        - Matrix Operating Environment
        - Network Automation
        - Openstack
        - Server Automation
        - SiteScope
        - uCMDB
        - vCenter
          - vCenter Clone Server
          - vCenter Custom Provider Pool Selection
            - Actions
              - vCenter Provider Pool Selection for Datastore
            - Subflows
          - vCenter Flex Server Count
          - vCenter Flex Server Resources
          - vCenter Modify Server Resources

Figure 57 shows the CSA action used in the vCenter design for performing custom provider and pool selection. The Process Definition field (see Figure 58) shows the details of the corresponding OO flow.

Figure 57. CSA Action for Custom Provider and Pool Selection



vCenter Provider Pool Selection for Datastore Properties ? ✕

Summary Properties

Action Properties

Configure the properties for this action below

Process Engine

OO-MACHINE-NAME

Process Definition

vCenter Provider Pool Selection for Datastore

/Library/CSA/3.2/Providers/vCenter/vCenter Custom Provider Pool Selec

Display Name *

vCenter Provider Pool Selection for Datastore

Description

Execution Order

2

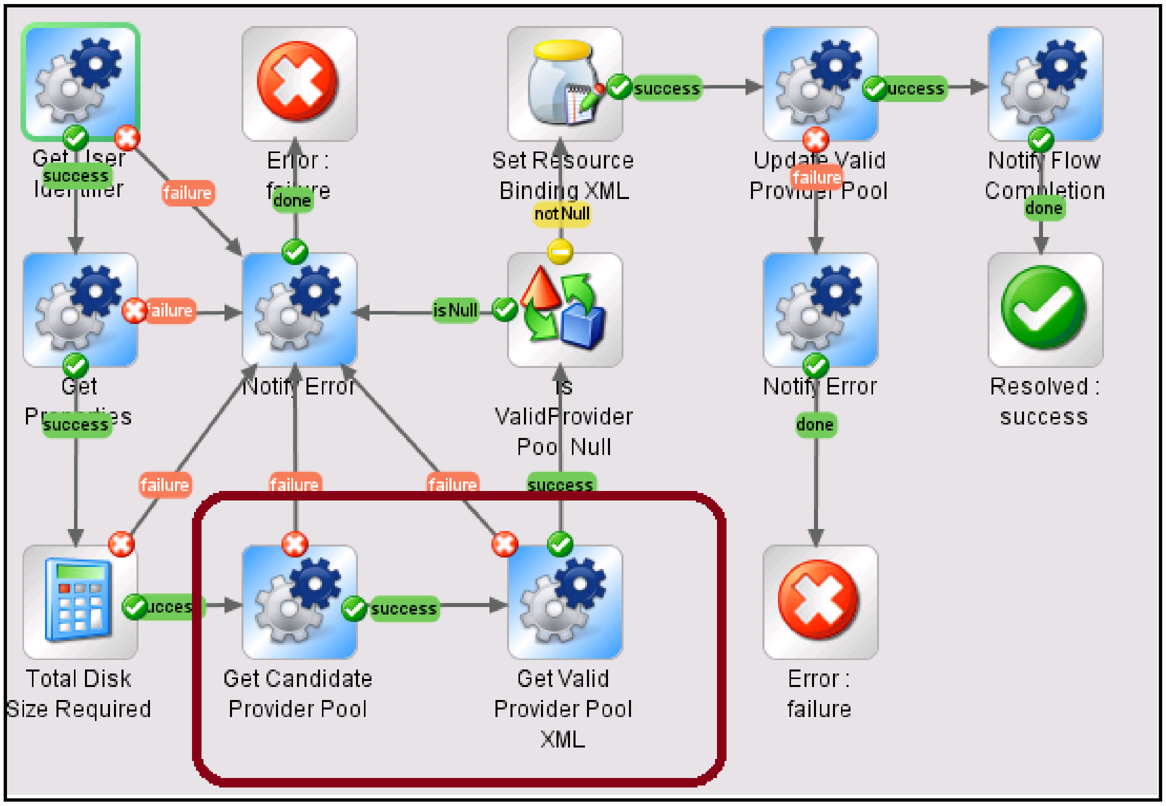
Execution Properties

☑ Fail on Error ☑ Error on Timeout

Save Reset Close

Figure 59 shows the flow that is invoked from CSA corresponding to the action shown in Figure 58. The steps which implement the refinement logic are shown in the red box below. For this flow, the refinement condition is based on the requesting disk size. A pool will be included in the list of valid pools only if required disk size criteria are satisfied.

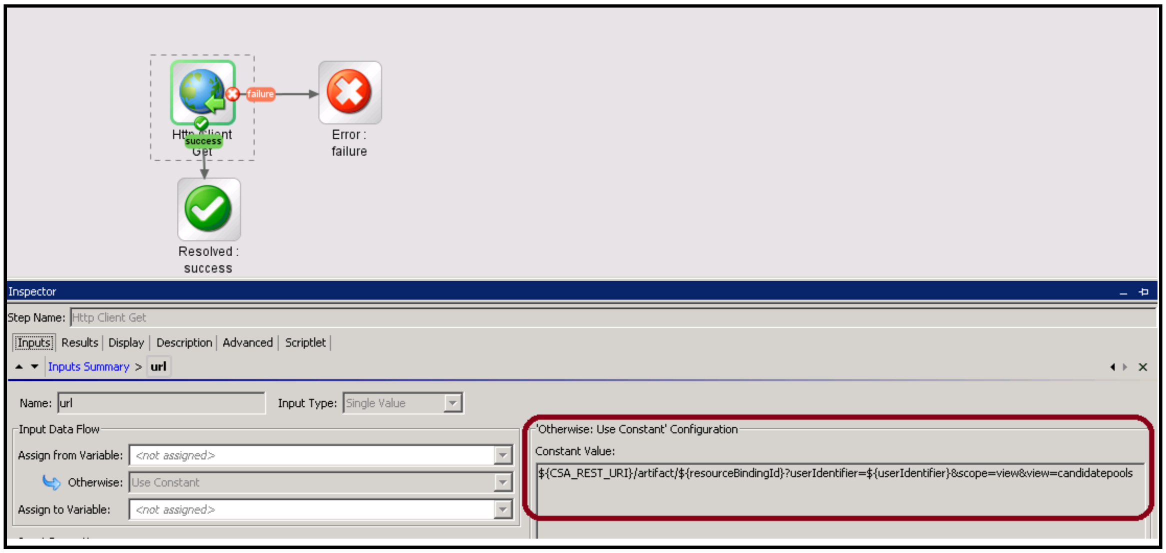
Figure 58. vCenter Provider Pool Selection for Datastore



Get Candidate Provider Pool

This step will retrieve all the candidate provider and pools from the CSA using the REST API call shown in the red box in Figure 60.

Figure 59. Candidate Pool REST Call



Step Name: Http Client Get

Inputs | Results | Display | Description | Advanced | Scriptlet

Inputs Summary > url

Name: url Input Type: Single Value

Input Data Flow

Assign from Variable: <not assigned>

Otherwise: Use Constant

Assign to Variable: <not assigned>

'Otherwise: Use Constant' Configuration

Constant Value:

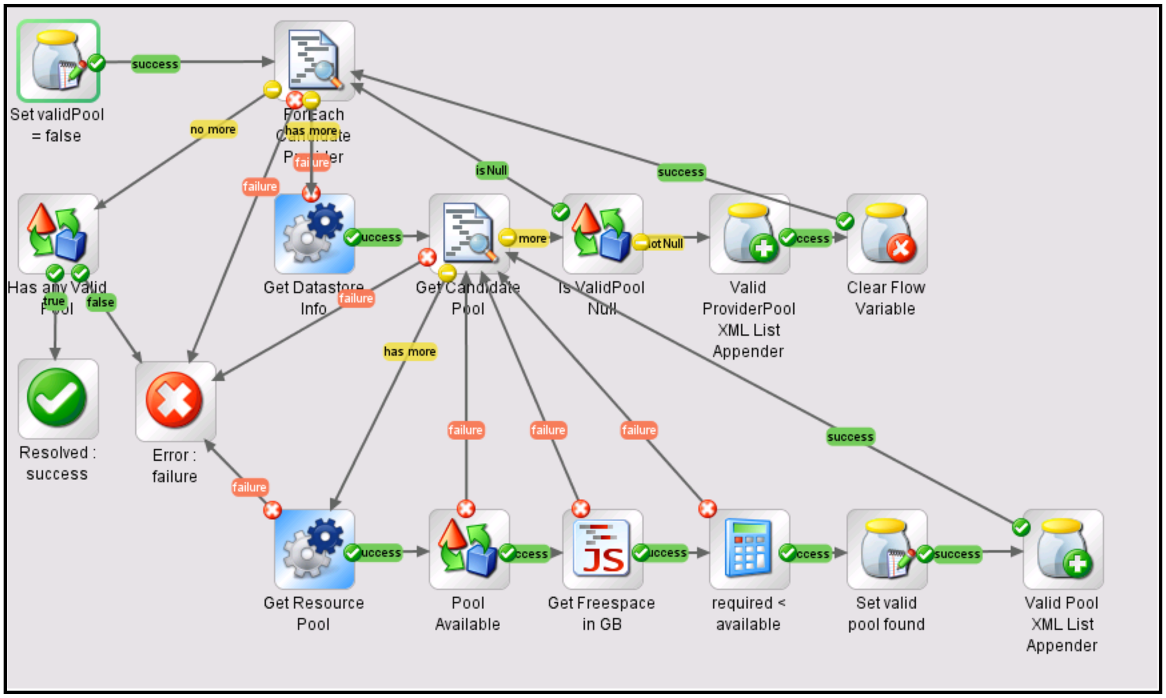
```
${CSA_REST_URI}/artifact/${resourceBindingId}?userIdentifier=${userIdentifier}&scope=view&view=candidatepools
```

Get valid provider pool XML

This step will browse through the candidate providers and pools retrieved in last step and filter the pool IDs based on storage. The subflow corresponding to this step is as show in Figure 61 and has following major steps:

- *Get Datastore Info* – Retrieves the list of data store names and free space available from the vCenter provider.
- *Get Resource Pool* – Retrieves the pool details like pool name, available, etc.
- *Valid Pool List Appender* - Appends or creates the valid pool IDs if the disk size requested is less than the free space available for a data store.
- *Valid Provider Pool List Appender* - Appends the provider ID to the valid pool id list created in the previous step, *Valid Pool List Appender.*

Figure 60. Create Valid Provider and Pool list



Appendix E: REST API for resource providers and pools

Get the resource pool details

Description: Get the resource pool artifact details

URI	`/artifact/<resourcePoolID>`
Method	**GET**
Parameters	userIdentifier=<user_id> Required; the user ID you want to use as credentials for this API call. This user should be a consumer user who has the necessary permissions for the data you want to work with.
	scope=[base|baseplusone|subtree] Optional; default is *base*. If value is *base*, then the object is returned. If value is *baseplusone*, then the object and its first level children are returned. If value is *subtree*, then the object and all of its descendants are returned.
	detail=[required|basic|standard|template|full] Optional; default is *full*. Some API calls do not support all possible values for this parameter.
Returns	200 - Updated 404 - Not found 500 - Server exception

Update resource pool details

Description: Artifact REST call to update availableToCsa or usedByCsa values

URI	`/artifact/<resourcePoolID>`
Method	**PUT**

Parameters	userIdentifier=<user_id>
	Required; the user ID you want to use as credentials for this API call. This user should be a consumer user who has the necessary permissions for the data you want to work with.
	_action_=merge

XML Input without Synchronize action	`<ResourcePool>`
	`    <name>${name}</name>`
	`    <displayName>${displayName}</displayName>`
	`    <poolReference>${poolReference}</poolReference>`
	`    <resourceProvider><id>${providerId}</id></resourceProvider>`
	`    <state>`
	`        <name>ACTIVE</name>`
	`    </state>`
	`    <artifactType>`
	`        <name>RESOURCE_POOL</name>`
	`    </artifactType>`
	`    <resourceCapacity>`
	`        <id>${capacityId}</id>`
	`        <resourceType>`

```
               <id>${resourceTypeId}</id>
               <name>${resourceTypeName}</name>
          </resourceType>
          <unit>
               <name>${unit}</name>
          </unit>
          <availabilityIndicator>
               <name>${availabilityIndicator}</name>
          </availabilityIndicator>
          <availableToCsa>${availableValue}</availableToCsa>
          <usedByCsa>${usedByCSA}</usedByCsa>
     </resourceCapacity>
</ResourcePool>
```

XML Input with Synchronize action

```
<ResourcePool>
     <name>${name}</name>
     <displayName>${displayName}</displayName>
     <poolReference>${poolReference}</poolReference>
     <resourceProvider><id>${providerId}</id></resourceProvider>
     <action>
          <id>${actionId}</id>
          <name>${actionName}</name>
          <processDefinition>
               <id>${processDefId}</id>
               <name>${actionName}</name>
          </processDefinition>
     </action>
     <state>
          <name>ACTIVE</name>
     </state>
     <artifactType>
          <name>RESOURCE_POOL</name>
     </artifactType>
     <resourceCapacity>
          <id>${capacityId}</id>
          <resourceType>
               <id>${resourceTypeId}</id>
               <name>${resourceTypeName}</name>
          </resourceType>
          <unit>
               <name>${unit}</name>
          </unit>
          <availabilityIndicator>
               <name>${availabilityIndicator}</name>
          </availabilityIndicator>
          <availableToCsa>${availableValue}</availableToCsa>
          <usedByCsa>${usedByCSA}</usedByCsa>
     </resourceCapacity>
</ResourcePool>
```

Appendix F: CSA Concepts

The following are relevant concepts for this document. For a comprehensive list of concepts, refer to the *CSA Concepts Guide*.

Artifact

A CSA model object that contains the necessary information to create and manage top-level model elements and their relationships.

Cloud Service Management Console

Software that provides a CSA design and administration interface.

Cloud Service

An entity for the delivery of cloud-computing capability to customers that can employ any of the following service models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), or Software-as-a-Service (SaaS).

Consumer Organization

An organization composed of CSA subscribers (or consumers). Consumer organizations provide enterprise-ready access to CSA cloud services. Members of a consumer organization place cloud service requests from service catalogs assigned to their organization. Consumer Service Administrators or CSA Administrators use the Cloud Service Management Console to set up and maintain consumer organizations.

Content

Programming entities such as OO flows and actions or CSA resource offerings and service designs. Content is imported into running instances of CSA and the configured OO process engine to drive functionality.

Custom Properties

User defined values configured on a service component, resource offering, or resource provider, typically read by OO flows during service provisioning. Custom properties can also be used to pass values between components of a service design and are used in conjunction with subscriber options to set properties on a service instance based on subscriber option selections. Certain properties can be made visible and/or editable in the Marketplace Portal.

HPE Cloud Service Automation (CSA)

A unique platform that orchestrates the deployment of infrastructure resources and complex multi-tier application architectures. CSA integrates and leverages the strengths of several HPE datacenter management and automation products, adding resource management, service offering design, and a customer portal to create a comprehensive service automation solution.

HPE Operations Orchestration (OO)

A software product that coordinates communication between integrated products and managed devices. Customized OO flows are essential to implementing the CSA service lifecycle.

Lifecycle

The stages of programmatically deploying a cloud service: initializing, reserving, deploying. Or conversely, the stages of removing a cloud service from deployment: *un*-deploying, *un*-reserving, and *un*-initializing. The service lifecycle also has a separate modification state.

Lifecycle Action

A function that is run automatically at a specified lifecycle state and sub-state. Lifecycle actions reference internal or external process definitions, which perform the specified action, such as initializing, reserving, or deploying a service subscription. Lifecycle actions can be applied to service components or resource offerings as part of the service lifecycle. Lifecycle actions can also be configured on stable states and made visible to subscribers in the Marketplace Portal.

Operations Orchestration (OO) Flow

A run-book automation workflow composed of operations, subflows, and integrations which implement a discrete action. Flows are synchronized with CSA, and presented as actions which can be configured with Resource Offerings or directly attached to components. OO flows are created, modified, and saved using HPE Operations Orchestration Studio. CSA includes a set of sample OO flows used by CSA's sample service designs.

Organization

An entity defined by the CSA Administrator, who determines a member's entry point into the cloud system and associates its members with services and resources. An organization can be a company, business unit, department, or group. Membership in an organization is determined by the organization's LDAP configuration, which CSA accesses to authenticate the user's login credentials.

Process Definition

A configuration that runs a specified internal (CSA) or external (OO flow) action.

Resource

A specific instance of software or infrastructure used to enable cloud service delivery.

Resource Binding

A link in a service design between a resource offering and a service component. For example, a resource offering for a specific VMware vCenter VM template can be linked to a Server Group service component. The resource binding ensures that the resource offering is provisioned as part of the service component deployment.

Resource Offering

A capability offered by a provider (or a group of providers) associated with a service design. Resource offerings are defined in the Cloud Service Management Console. An offering has a single provider type and a single resource category. An offering is associated with providers to indicate which providers support the offering.

Resource Provider

A management platform that provides either Infrastructure-as-a-Service (IaaS) or Software-as-a-Service (SaaS) to the cloud. For example, a provider of Matrix OE services provisions infrastructure and basic applications, while a provider of SiteScope services monitors applications.

Resource Environment

A mechanism for grouping related resource providers. One or more resource environments can be linked to a service catalog to restrict provider selection at subscription time. When provider selection occurs during service provisioning, only providers belonging to one or more of the environments associated with the service catalog will be eligible for selection.

Service Business Manager

A CSA user role. The Service Business Manager creates and manages service offerings and service catalogs.

Service Catalog

A collection of service offerings, configured in the Cloud Service Management Console. Subscribers see service offerings from organization-specific catalogs when they log into the Marketplace Portal.

Service Component

Represents one element required to realize a service subscription and provides a framework to describe the actions and resource offerings required to realize, manage, and retire this element.

Service Design

A template (or blueprint) for an orderable service. A service design includes a hierarchy of service components, plus resource bindings, subscriber options, lifecycle actions, and custom properties, as defined by the Service Designer.

Service Designer

A CSA user role. The Service Designer designs, implements, and maintains service designs (also referred to as blueprints).

Service Offering

An entity developed by the Service Business Manager to refine existing service designs (or blueprints) and then publish them to a service catalog. A service offering adds pricing, images, and other specific information required for the subscription process.

Subscription

An instance of a service offering as requested by a subscriber and granted through the relevant approval process. Subscriptions incur costs according to a cost structure developed by the Service Business Manager.

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

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.

Go to the HPE Software Support website and sign-in or register to check for recent updates or to verify that you are using the most recent edition of a document. 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.